

求解 TSP 问题的离散狼群算法

吴虎胜¹, 张凤鸣^{2a}, 李浩^{2a}, 梁晓龙^{2b}

(1. 武警工程大学 装备工程学院, 西安 710086; 2. 空军工程大学

a. 装备管理与安全工程学院, b. 空管领航学院, 西安 710051)

摘要: 通过定义反转算子, 对人工狼位置和智能行为重新进行整数编码设计, 并结合概率近邻初始化方法, 提出一种求解旅行商问题的离散狼群算法. 该算法保留了狼群算法基于职责分工的协作式搜索特性, 并较好地平衡了算法的广度开拓和深度开采能力. 采用 C-TSP 问题和 TSPLIB 数据库中的多组 TSP 问题作为实验用算例, 并将所提出算法与其他 5 种智能优化算法进行对比, 仿真结果表明, 所提出算法在求解准确率、稳定性和所需迭代次数等方面具有相对优势.

关键词: 进化计算; 群体智能; 离散狼群算法; 组合优化; 旅行商问题

中图分类号: TP18; TP393

文献标志码: A

Discrete wolf pack algorithm for traveling salesman problem

WU Hu-sheng¹, ZHANG Feng-ming^{2a}, LI Hao^{2a}, LIANG Xiao-long^{2b}

(1. Materiel Engineering College, Armed Police Force Engineering University, Xi'an 710086, China; 2a. Materiel Management and Safety Engineering College, 2b. Air Traffic Control and Navigation College, Air Force Engineering University, Xi'an 710051, China. Correspondent: WU Hu-sheng, E-mail: wuhusheng0421@163.com)

Abstract: Based on designing of reverse operator integer coding, the artificial wolves' position and intelligent behaviors are redesigned. Then combined with the probabilistic nearest neighbor method, a discrete wolf pack algorithm(DWPA) is proposed to solve traveling salesman problem(TSP). The DWPA preserves the cooperative searching feature based on job distribution of the wolf pack, and makes a good balance between the searching ability in breadth and depth. Algorithm efficiency of TSP problems against five existing algorithms is examined for C-TSP and some TSPLIB instances of various sizes. Simulation results show that the DWPA has more advantages on accurate rate, computational robustness and iteration number.

Keywords: evolutionary computation; swarm intelligence; discrete wolf pack algorithm; combinatorial optimization; traveling salesman problem

0 引言

旅行商问题(TSP)是经典的组合优化问题, 问题旨在求得一条遍历所有城市的最短回路. TSP 问题应用十分广泛, 很多现实问题都可以抽象为该问题进行求解, 如: 无人机航迹规划、网络路由器布设、车辆调度、计算机联网、机器人控制等^[1]. 研究 TSP 问题具有重要的理论和实际价值, 但 TSP 问题属于 NP 难问题, 求解较为困难. 诸如分支界定等传统求解方法无法应对因城市数量逐渐增加所引发的“组合爆炸”问题.

自然界大量的生物进化与群体协作为学者们求

解复杂优化问题提供了很多启示^[2]. 遗传算法^[3]、粒子群优化算法^[4]、蚁群算法^[5]、人工蜂群算法^[6]、萤火虫群优化算法^[7]、生物地理迁移算法^[8]、帝国主义竞争算法^[9]等智能算法被用于解决 TSP 问题. 这些智能优化算法大大缩小了解空间并增加了在有限时间内找到问题最优解的概率. 但很多算法同时也存在不足, 如收敛速度慢和陷入局部极值等^[10]. 因此, TSP 问题虽早已提出, 但仍被广大学者视为非常具有研究价值和难度的问题之一^[11]. 同时, TSP 问题也是评价算法离散空间寻优性能的一项有效的准则. 启发于生物狼群的群体生存智慧, 文献[12]从头狼、探狼、猛狼的

收稿日期: 2014-07-03; 修回日期: 2014-12-29.

基金项目: 国家自然科学基金项目(61472442, 61472443, 61203268); 武警工程大学理论项目(WJY201511, JLX201540, JLX201503).

作者简介: 吴虎胜(1986—), 男, 讲师, 从事进化计算、信息系统工程与智能决策、智能数据挖掘的研究; 张凤鸣(1963—), 男, 教授, 博士生导师, 从事信息系统工程与智能决策、智能数据挖掘等研究.

不同职责分工的角度分析狼群捕猎行为,设计了一种新的群体智能算法——狼群算法(WPA).文献[13]在此基础上,引入0-1二进制编码,并基于运动算子提出了二进制狼群算法,成功用于0-1高维背包问题的求解.本文在WPA算法的基础上,引入整数序列编码,并结合概率近邻初始化方法,提出一种离散狼群算法(DWPA)用于求解TSP问题,并采用多组不同规模的算例和多种智能优化算法的对比仿真验证了所提出算法的有效性.

1 离散狼群算法

1.1 一些定义

设在 $n \times m$ 的解空间中 n 为狼群规模, m 为变量数对应TSP问题中的城市数量.人工狼 i 的位置定义为 $X_i = \{x_{i1}, x_{i2}, \dots, x_{im}\}$,人工狼所嗅到的猎物气味浓度为 $Y = f(X)$; X_i 对应TSP问题中的一条可行路径编码, Y 对应于路径长度,其中 x_{ij} 为第 i 条可行路径编码的第 j 个编码位置的城市.另有 $x_{ip} \neq x_{iq}$, $p \neq q$,且 $x_{ip}, x_{iq}, p, q, j \in \{1, 2, \dots, m\}$.

定义1(人工狼之间的距离) 人工狼 p 与 q 之间的距离定义为两条路径编码 X_p 与 X_q 之间的最长公共子序列(LCSS)距离,LCSS距离具体求解方法参见文献[14].

定义2(近邻度) 近邻度 $\text{Near}(b \rightarrow a) = j$ 表示城市 b 是距离 a 第 j 近的城市.如 $\text{Near}(2 \rightarrow 5) = 1$ 就表示城市2是距城市5最近的城市;同理,若城市1是距城市4第3近的城市,则城市1相对于城市4的近邻度 $\text{Near}(1 \rightarrow 4) = 3$.

定义3(近邻概率) 近邻概率 $P(c \rightarrow d)$ 表示在确定旅行路径时,经过城市 c 后,城市 d 作为下个旅行城市的概率.

定义4(反转算子) 对于某条路径,反转算子 $\text{Rev}(X_i, a, b)$, $a \neq b$,表示将路径 X_i 中 a 与 b 之间(包括 a 和 b)所有的城市进行逆序重排.如果 $X_i = [1, 3, 6, 4, 2, 5, 7, 8]$, $a = 3$, $b = 5$,则 $\text{Rev}(X_i, a, b) = [1, 5, 2, 4, 6, 3, 7, 8]$.

1.2 概率近邻初始化

参考大量文献和TSPLIB提供的TSP问题的最短路径图不难发现,TSP问题的最优解都有两个基本特点:1)各城市总是与其距离较近的近邻城市相连;2)最优路径中没有交叉.因此,设计初始解的概率近邻初始化方法,具体步骤如下:

1) 求取近邻城市矩阵 A .对于某一目标城市 i ,求取其他城市相对于 i 的近邻度,接近邻度的升序排列其他城市. $A(i, j)$ 表示相对城市 i 近邻度为 j 的城市,即有

$$\text{Near}(A(i, j) \rightarrow i) = j.$$

其中: $A(i, m) = i, j \in \{1, 2, \dots, m\}$ 且 $A(i, j)$ 一般不等于 $A(j, i)$.若 $i = 2, A(i, :) = [1, 4, 5, 3, 2], j = 3$,则 $A(2, 3) = 5$ 表示距城市2第3近的城市为5,同时有 $\text{Near}(5 \rightarrow 2) = 3$.

2) 求取近邻概率矩阵 B .对于某条路径, $B(i, j)$ 表示若城市 i 为当前旅行城市,则与 i 近邻度为 j 的城市 $A(i, j)$ 成为下个旅行城市的概率,即 $B(i, j) = P(i \rightarrow A(i, j))$.规定 $B(i, m) = 0$,因为合法的路径中某个城市经过且仅经过一次.若 $i = 2, B(i, :) = [0.4, 0.3, 0.2, 0.08, 0.02], j = 3$,则 $B(i, j) = 0.2$ 表示经过城市2后接着旅行城市 $A(2, 3) = 5$ 的概率为0.2,即有 $P(2 \rightarrow 5) = 0.2$.

若 D 是两两城市间距离组成的矩阵,则 $D(p, q)$ 表示城市 p 与城市 q 间的距离.其中 $p = q$ 时, $D(p, q) = \infty, 1/D(p, q) = 0$.与上面矩阵 A 和 B 对应, $p = i, q = A(i, j)$,则可得到过渡矩阵

$$B^*(i, j) = \frac{1}{\sum_{t=1}^m \frac{1}{D(p, t)}} = \frac{1}{\sum_{t=1}^m \frac{1}{D(i, A(i, t))}}. \quad (1)$$

据已公布的实验统计,目前所采用实例的最优路径中相邻城市一般彼此都在距其距离较近的前20个城市之内^[15].因此,这里取最大近邻度为 $N_{\max} = 20$, N_{\max} 表示相对于某城市 i ,近邻度大于 N_{\max} 的城市将以很小的概率 p_0 成为 i 的下个旅行城市,一般取 $p_0 = 0.01$.因此,可由过渡矩阵 B^* 计算得到近邻概率矩阵 B ,如下式所示:

$$B(i, j) = \begin{cases} (1 - p_0) \frac{B^*(i, j)}{\sum_{j=1}^{N_{\max}} (B^*(i, j))}, & j \leq N_{\max}; \\ p_0 \frac{B^*(i, j)}{\sum_{j=N_{\max}+1}^m (B^*(i, j))}, & j > N_{\max}. \end{cases} \quad (2)$$

3) 随机选取一个城市 i 作为出发城市,由近邻概率矩阵 B 确定其下个城市与之的近邻度 j ,据近邻城市矩阵 A 确定 i 的下个旅行城市 $A(i, j)$;再以 $A(i, j)$ 作为出发城市重复前述过程确定其下个旅行城市,如此直至旅行完所有的城市即生成一条合法路径.这样可以生成 n 条合法路径,进而完成初始解的概率近邻初始化.需说明的是,为防止生成的路径中重复出现相同的城市,在依据近邻概率确定城市 i 的下个城市前,须先将已旅行过城市的对应近邻概率置为0.

另外,由于严格区分了近邻度 $\text{Near}(b \rightarrow a)$ 和 $\text{Near}(a \rightarrow b)$,近邻城市矩阵 A 和近邻概率矩阵 B 同样满足非对称性.因此,概率近邻初始化方法原则上

也适用于非对称 TSP 问题初始解的生成。

1.3 智能行为描述

DWPA 中头狼产生规则和狼群更新机制同文献 [12], 下面将详述游走、召唤和围攻 3 种智能行为。

1) 游走行为. 将除头狼外的所有人工狼视为探狼在解空间中搜索猎物. 设 Y_i 和 Y_{lead} 分别为探狼 i 和头狼所感知的猎物气味浓度, 若 $Y_i > Y_{lead}$, 则探狼 i 替代头狼并发起召唤行为; 若 $Y_i \leq Y_{lead}$, 则探狼分别向 h 个方向前进一步, 并记录每次所感知的猎物气味浓度. 具体地, 向第 p 个方向 ($p = 1, 2, \dots, h$) 前进一步即是在探狼 i 的编码 $X_i = \{x_{i1}, x_{i2}, \dots, x_{im}\}$ 中随机选取一个城市 x_{ij} , 依近邻概率矩阵 B 和近邻矩阵 A 确定城市 x_{ij} 的近邻城市 x_{ik}, x_{ij+1} 为 X_i 中 x_{ij} 的下一个编码位城市, 则执行反转算子 $Rev(X_i, x_{ij+1}, x_{ik})$ 一次. 设 Y_{i0} 和 Y_{ip} 分别为其向第 p 个方向前进前、后所感知的气味浓度, 则选择满足下式的方向前进一步, 更新探狼的位置 X_i .

$$\{p|Y_i^* = \max\{Y_{ip}, p = 1, 2, \dots, m\}, Y_i^* > Y_{i0}\}. \quad (3)$$

重复以上的游走行为, 直到 $Y_i > Y_{lead}$ 或游走次数 T 达到最大游走次数 T_{max} .

2) 召唤行为. 头狼通过嚎叫召集猛狼向头狼位置 X_s 聚集. 将除头狼外所有人工狼视为猛狼, 猛狼快速向头狼所在位置奔袭. 具体地, 在头狼位置编码中随机选取长度为 $step_b$ 的一段, 替换猛狼 X_i 中以相同开头或结尾、长度也为 $step_b$ 的一段, 并将其他城市做相应调整, 使得所有城市出现且仅出现一次. 如 $X_s = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$, $X_i = [9, 6, 4, 3, 1, 7, 2, 10, 8, 5]$, $step_b = 3$, 随机选取 X_s 中的一段 $[3, 4, 5]$ 替换 X_i 中的 $[3, 1, 7]$, 得到 $X_i = [9, 6, 4, 3, 4, 5, 2, 10, 8, 5]$, 然后依据映射关系 $4 \rightarrow 1, 5 \rightarrow 7$ 进行调整, 得到 $X_i = [9, 6, 1, 3, 4, 5, 2, 10, 8, 7]$. 这种编码的调整对猛狼大部分的编码值进行了保留, 体现了猛狼 i 的个体经验; 将头狼编码位值直接替换猛狼原有的编码值, 体现了群体优秀个体 (头狼) 的经验对个体行为的指导.

奔袭途中, 若 $Y_i > Y_{lead}$, 则该猛狼成为头狼并发起召唤行为; 若 $Y_i \leq Y_{lead}$, 则猛狼 i 继续奔袭直到 $d_{is} \leq d_{near}$, 其中 d_{is} 为猛狼 i 与头狼间的距离, 判定距离 $d_{near} = \lceil m/w \rceil$, $\lceil * \rceil$ 为向上取整, w 为距离判定因子.

3) 围攻行为. 头狼指挥猛狼和探狼对猎物进行围攻, 将头狼的位置 X_s 视为猎物所在位置. 首先在参与围攻的人工狼 i 的编码 $X_i = \{x_{i1}, x_{i2}, \dots, x_{im}\}$ 中随机产生一个城市 a , 查找猎物位置 $X_s = \{x_{d1}, x_{d2}, \dots, a, b, \dots, x_{dm}\}$ 中 a 的下一个城市为 b ; 然后在 X_i 中将城市 b 插入到城市 a 之后. 由于参与围攻的

人工狼相互之间的距离较近, 为尽快将猎物捕获, 都会进行信息共享和交流. 具体地, 将两两人工狼的位置编码进行比较操作, 即首先在参与围攻的人工狼 i 的编码 $X_i = \{x_{i1}, x_{i2}, \dots, x_{im}\}$ 中随机产生一个城市 c 作为一段城市序列的起始城市, 并确定一段长度为 $step_c$ 的城市序列段 s_1 ; 在另一只人工狼的位置编码 $X_j = \{x_{j1}, x_{j2}, \dots, x_{jm}\}$ 中查看是否存在包含相同城市但顺序不同的城市段 s_2 , 若存在, 则比较 s_1 和 s_2 的路径长度, 若 $length(s_1) < length(s_2)$, 则利用 s_1 替换 X_j 中的 s_2 , 反之亦然.

最后对实施围攻行为的人工狼位置进行贪婪决策, 决定是否对其位置进行改变.

围攻行为中前者的插入操作体现了狼群的信息传递与优秀个体 (头狼) 的经验共享机制; 后者的对比操作可理解为围攻时人工狼之间的高度信息交流和狼群在猎物周围小范围内调整以达到最佳攻击位置, 体现了算法在优秀解域附近对最优解的精细搜索, 同时这种带有随机性的位置调整也有利于减小算法陷入局部极值的概率.

综上, DWPA 算法的计算流程如图 1 所示.

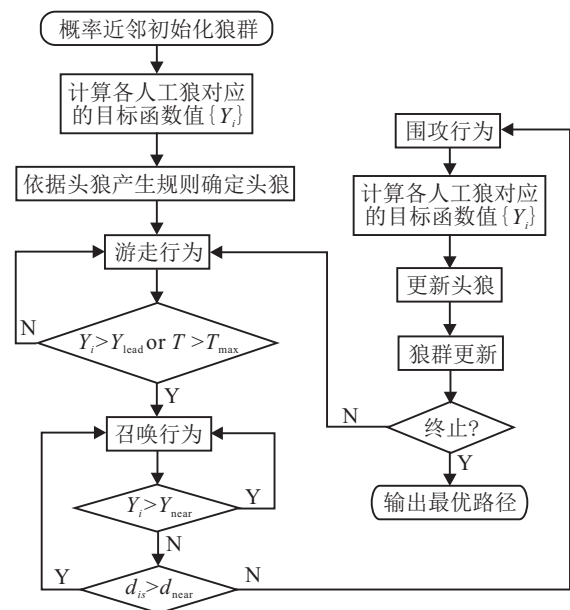


图 1 DWPA 的算法流程

2 实验与分析

为了验证算法性能, 首先求解中国旅行商 (C-TSP) 问题以验证所提出算法的有效性. 为进一步对比分析, 采用 TSPLIB 的标准库的多组数据, 并与蚁群算法、遗传算法等多种智能算法进行对比计算. 实验环境同文献 [12].

2.1 算法有效性测试, C-TSP 问题

C-TSP 问题即是要进行“走遍中国”的旅游, 问如何规划旅游路径才能游遍所有的直辖市、省会和自治

区首府(不含港澳)且路线既不重复又最短. 如按照枚举法, 约有 1.325×10^{32} 个方案, 数量庞大, 这里采用上述的离散狼群算法进行 50 次独立计算. 图 2 为计算得到的最优路径图.

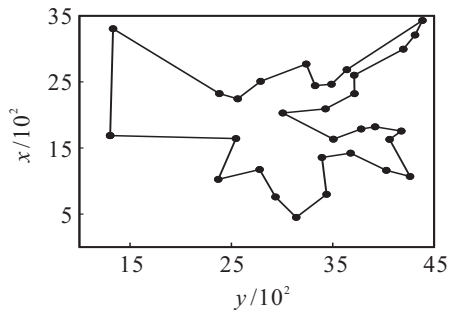


图 2 DWPA 计算 C-TSP 问题所得的最优路径

图 3 是 50 次独立计算的迭代收敛曲线. 如图 3 所示, 算法收敛速度较快, 平均迭代次数为 26 次. 50 次独立计算中, DWPA 有 45 次得到最优解 15 377.71, 且这一结果优于其他文献所得出的 15 379.16^[16]、15 942^[17]、15 381^[18] 和 15 404^[19] 的结果. 统计 50 次计算的结果, DWPA 算法所得 C-TSP 问题的平均解为 15 378.43, 最差解 15 397.43 的偏差为 0.13%. 由此可见 DWPA 具有较好的求解精度和求解稳定性.

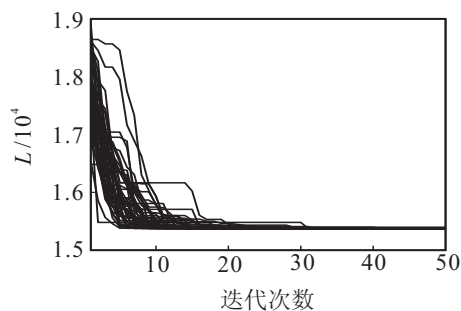


图 3 求解 C-TSP 问题 50 次独立计算的迭代曲线

2.2 算法性能比较

为进一步比较各算法性能, 将离散狼群算法 (DWPA) 与遗传算法 (GA)^[3]、蚁群算法 (ACO)^[5]、模拟退火算法 (BSA)^[20]、改进模拟退火算法 (ISA)^[21] 和交叉粒子群算法 (CPSO)^[22] 就解决不同规模的 TSP 问题进行对比计算. 采用 TSPLIB 标准库的 bays29、eil51、berlin52、gr96、pr144 和 gr202 共 6 个 T-SP 问题进行多次寻优计算. 对于实验结果, 引入如下评价标准: 最优值 best, 最差值 worst, 平均值 ave, 标准差 std, 成功率 SR, 偏差 PDbest, 平均迭代收敛次数 T_{ave} . 设 S_0 为 TSPLIB 提供的最短路径值, 若计算得到最优路径值满足 $best/S_0 < 1.03$, 则视此次寻优计算成功^[23]. 另外偏差 PDbest 可由下式计算得到:

$$PDbest = \frac{\sum_{k=1}^T (best_k - s_0)}{T s_0} \times 100\%. \quad (4)$$

其中: $best_k$ 为第 k 次独立计算所得 TSP 问题的最短路径长度, T 为独立计算的次数, 取 $T = 30$. PDbest 表示算法所求结果相对于 TSPLIB 的标准库给出解的偏差, 一般为正, 若 PDbest 为负数则说明算法求得最短路径比 TSPLIB 给出的更短, 且绝对值越大表示算法求得结果越好.

实验设置最大迭代次数 iter_max 除 gr202 为 3 000 外, 其他皆为 1 000. 各算法初始种群规模皆为 100, 其他参数依据相关文献的基本原则设置如下:

BSA 和 ISA 算法的初始温度 $T_0 = 1 000$, 终止温度 $T_{end} = 1e-3$, 各温度下的迭代次数为 $L = 500$, 降温速率 $q = 0.986 3$;

ACO 算法的信息素重要程度因子 $\alpha = 1$, 启发式因子权重 $\beta = 5$, 信息素挥发因子 $\rho = 0.1$, 常系数 $Q = 1$;

CPSO 算法中学习因子 $c_1 = c_2 = 2$, 惯性权重 $= 0.729 8$, 个体速度限制为 $[-0.5, 0.5]$;

DWPA 算法中最大游走次数 $T_{max} = 10$, 距离判定因子 $w = 18$, 步长 $step_b = 6$, $step_c = 18$, 更新比例因子 $\beta = 2$;

GA 算法交叉和变异概率分别为 $p_c = 0.9$ 和 $p_m = 0.05$, 采用随机多点交叉和变异.

运用 ACO、CPSO、GA、BSA、ISA 和 DWPA 六种方法对 bays29、eil51 等 6 个 TSP 问题分别独立运算 30 次后统计实验结果如表 1 所示, 其中算法迭代终止条件为: 达到最大迭代次数或最优解在 100 次迭代内无更新. 由于 TSP 问题名称中已包含问题的维数信息 (如 bays29 的维数为 29, eil51 的维数为 51 等), 故此项在表 1 中省略. 为了直观认识, 图 4 给出了 DWPA 求解 6 个 TSP 问题所得的最优路径, 横纵坐标皆分别表示 X 坐标和 Y 坐标.

表 1 中, 由最优解和平均值的对比可知, DWPA 对于不同规模的 TSP 问题均能搜索到最优路径, 其中 bays29、eil51、gr96、pr144 和 gr202 这 5 个 TSP 问题计算得到的最优路径都比 TSPLIB 提供的最优路径更优, 特别对于 eil51、gr96、pr144、gr202 这 4 个 TSP 问题, 30 次计算所得平均值都要优于其他 5 种算法的最优值, 充分反应了 DWPA 算法具有较高的寻优精度; 由最差解、标准差的对比可知, 虽然 TSP 问题的维数不唯一, 但除 pr144 外, 对于其他 5 个问题, DWPA 算法的标准差都较小, 显示算法具有较好的计算鲁棒性. 成功率、偏差和平均迭代次数体现了算法收敛到某一精度最优解的能力, 如表 1 所示, DWPA 算法对 6 个 TSP 问题的 30 次计算的成功率都为 100%, 偏差皆小于 1%, 平均迭代次数在 450 次以内, 体现了 DWPA 具有良好的全局收敛能力, 而其他 5 种算法都

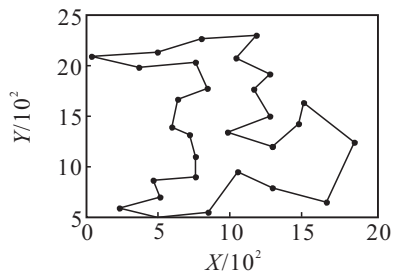
表 1 6 种算法求解不同 TSP 问题时的性能比较

TSP 问题	TSPLIB 最优解	算法	best	worst	ave	std	SR/%	PDbest/%	T_{ave}
bays29	9.293 3e+3	BSA	9.186 8e+3	1.093 0e+4	9.727 6e+3	353.982 7	36.67	4.692 3	33
		ACO	9.178 4e+3	9.318 6e+3	9.274 3e+3	42.101 9	100	-0.18	435
		GA	9.105 9e+3	9.777e+3	9.329 9e+3	165.911 6	86.67	0.42	344
		ISA	9.074 1e+3	9.520 9e+3	9.168 3e+3	111.272 6	100	-1.32	49
		CPSO	9.074 1e+3	9.598 0e+3	9.243 8e+3	185.420 8	93.33	-0.51	522
		DWPA	9.074 1e+3	9.104 4e+3	9.078 6e+3	8.407 4	100	-2.29	12
eil51	429.983 3	BSA	458.069 9	596.487 3	512.504 4	29.217 4	0	19.19	79
		ACO	442.030 0	457.709 9	451.097 9	4.550 7	6.67	4.91	460
		GA	437.340 8	479.245 7	455.921 4	9.426 8	3.33	6.02	394
		ISA	436.667 3	462.125 9	448.730 6	7.257 4	23.33	4.36	93
		CPSO	444.807 6	494.712 5	464.464 0	12.400 3	0	8.02	878
		DWPA	428.871 8	437.216 7	431.215 7	2.564 6	100	0.29	52
berlin52	7.545 4e+3	BSA	8.200 4e+3	1.005 7e+4	9.204 4e+3	445.006 3	0	22	58
		ACO	7.641 8e+3	7.847 4e+3	7.701 9e+3	51.505 6	86.67	2.09	455
		GA	7.606 9e+3	8.616 9e+3	8.156 3e+3	242.874 3	6.67	8.11	478
		ISA	7.544 4e+3	8.395 0e+3	7.986 6e+3	189.950 7	6.67	5.86	71
		CPSO	7.772 7e+3	8.694 1e+3	8.210 3e+3	226.124 3	0	8.83	882
		DWPA	7.545 4e+3	7.598 4e+3	7.546 2e+3	9.873 0	100	0.02	34
gr96	512.309 4	BSA	676.682 2	925.158 2	796.760 9	59.373 7	0	55.52	122
		ACO	546.889 3	564.370 7	553.474 4	4.475 9	0	8.04	574
		GA	522.968 7	597.826 8	563.747 7	16.408 2	3.33	10.04	974
		ISA	524.467 6	579.503 5	546.875 6	13.294 2	3.33	6.75	120
		CPSO	661.391 9	834.751 0	726.390 0	40.004 8	0	41.79	987
		DWPA	510.886 3	525.896 7	517.154 5	4.522 7	100	0.95	258
pr144	58 537	BSA	9.466 3e+4	1.289 3e+5	1.138 5e+5	9.254 7e+3	0	94.5	1000
		ACO	5.916 8e+4	5.998 2e+4	5.958 6e+4	189.124 2	100	1.79	512
		GA	8.372 3e+4	1.403 0e+5	1.077 8e+5	9.943 9e+3	0	84.12	999
		ISA	5.860 5e+4	6.257 2e+4	5.956 6e+4	1.058 8e+3	83.22	1.76	976
		CPSO	1.251 0e+5	1.763 2e+5	1.495 2e+5	1.436 3e+4	0	105.42	996
		DWPA	5.853 5e+4	5.877 9e+4	5.860 4e+4	81.888 1	100	0.1	245
gr202	549.998 1	BSA	636.862 6	719.430 9	669.868 6	20.069 8	0	21.79	923
		ACO	516.872 1	543.797 6	532.685 1	8.127 6	100	-3.15	1 343
		GA	520.582 7	556.750 0	535.771 8	8.409 0	100	-2.59	998
		ISA	501.209 2	521.273 4	511.878 2	5.048 5	100	-6.93	888
		CPSO	682.519 8	790.284 5	729.900 4	25.531 4	0	32.71	995
		DWPA	490.066 9	507.553 4	498.050 6	3.706 9	100	-9.45	404

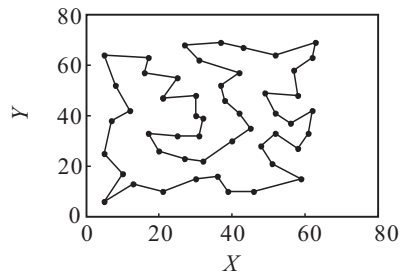
不同程度地陷入局部最优且需要更多的迭代次数才能收敛。

分析认为: 对于如 TSP 这样的结构复杂、具有庞大搜索空间的离散优化问题, GA 算法和 ACO 算法的搜索时间都较长, 容易出现停滞, 收敛速度慢、易陷入局部最优^[24]; BSA 算法和 CPSO 算法收敛速度较快, 但精度较低, 也易陷入局部最优^[25]; ISA 算法的性能有所改进, 但计算精度和稳定性方面仍劣于 DWPA 算法。而 DWPA 算法却不然, 首先初始解的概率近邻初始化提高了初始解质量; 游走行为一方面通过人工

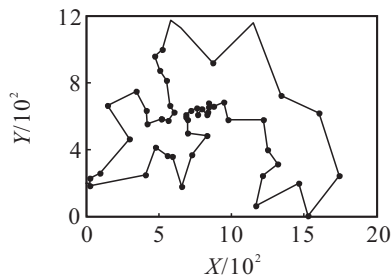
探狼向不同方向的嗅探, 使人工狼能充分遍历搜索解空间, 一方面通过依概率近邻原则执行大量的反转算子, 使得距离近的城市以较大概率相连, 同时也使得路径中不存在交叉的边, 最大限度地保证城市顶点构成凸多边形, 从而更好地构建最短回路; 召唤行为通过将头狼的随机长度的编码段复制给其他的狼, 使得狼群快速地向优良解区域移动, 进一步缩小了搜索范围; 而围攻行为中通过插入操作和人工狼在猎物周围所进行的两两信息交互, 即比较随机数量城市路段的长度, 实现了局部的深入搜索。DWPA 算法在狼群群



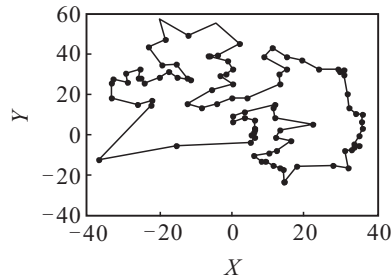
(a) bays29(length = 9 074.148)



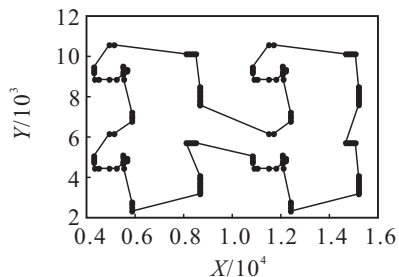
(b) eil51(length = 428.871 8)



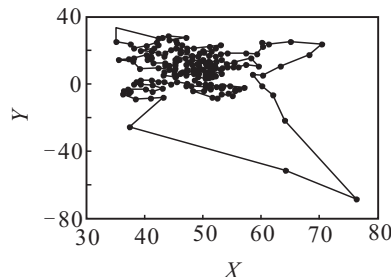
(c) berlin52(length = 7 544.365 9)



(d) gr96(length = 510.8863)



(e) pr144(length = 58 535.221 8)



(f) gr202(length = 490.066 9)

图4 DWPA计算得到的6个TSP问题的最优路径

体智能寻优机制中加入了局部搜索算法,较好地平衡了算法的广度开拓和深度开采能力.加之,DWPA算法很多参数和操作都带有大量的随机性,可更好地遍历解空间,这种导向式的随机搜索最终得到了如表1所示的较好的寻优效果.

3 结 论

本文基于狼群群体猎食智慧,通过依据TSP问题特点设计了概率近邻初始化方法,重新定义人工狼的位置和智能行为,构建了一种求解TSP问题的离散狼群算法.通过C-TSP问题和TSPLIB数据库的多个TSP问题的实验计算表明:相对于蚁群算法、遗传算法、交叉粒子群算法等经典算法,DWPA算法具有较好的求解精度和求解稳定性.DWPA算法的初始路径生成和对路径编码的变换(除反转算子外)都考虑到城市旅行的先后顺序和相互距离的不对称性,因此,DWPA算法只要稍加改变,理论上也适用于非对称TSP问题.今后将引入更多的TSP算例和非对称TSP问题算例进行数值实验和算法的参数分析,并扩大算法的实际应用领域,如无人机的航迹规划等.

参考文献(References)

- [1] 高海昌,冯博琴,朱利.智能优化算法求解TSP问题[J].控制与决策,2006,21(3):241-247.
(Gao H C, Feng B Q, Zhu L. Reviews of the meta-heuristic algorithms for TSP[J]. Control and Decision, 2006, 21(3): 241-247.)
- [2] 梁雅俊,肖明清,梁鹏,等.基于灰色局势决策模型的ATS资源选择与配置[J].空军工程大学学报:自然科学版,2015,16(5):77-80.
(Liang Y J, Xiao M Q, Liang P, et al. Test resource selection and configuration of automatics test system based on grey situation decision-making model[J]. J of Air Force Engineering University: Natural Science Edition, 2015, 16(5): 77-80.)
- [3] Merz P, Freisleben B. Genetic local search for the TSP: New results[C]. Proc of the 1997 IEEE Int Conf on Evolutionary Computation. Indianapolis: IEEE Press, 1997: 159-164.
- [4] Wang K P, Huang L, Zhou C G, et al. Particle swarm optimization for traveling salesman problem[C]. Proc of the Int Conf on Machine Learning and Cybernetics. Xi'an: IEEE Press, 2003: 1583-1585.
- [5] Dorigo M, Luca M G. Ant colonies for the traveling salesman problem[J]. BioSystems, 1997, 43(2): 73-81.
- [6] 胡中华,赵敏.基于人工蜂群算法的TSP仿真[J].北京理工大学学报,2009,29(11):978-982.
(Hu Z H, Zhao M. Simulation on traveling salesman problem(TSP) based on artificial bees colony algorithm[J].

- Trans of Beijing Institute of Technology, 2009, 29(11): 978-982.)
- [7] 周永权, 黄正新. 求解TSP的人工萤火虫群优化算法[J]. 控制与决策, 2012, 27(12): 1816-1821.
(Zhou Y Q, Huang Z X. Artificial glowworm swarm optimization algorithm for TSP[J]. Control and Decision, 2012, 27(12): 1816-1821.)
- [8] Mo H W, Xu L F. Biogeography migration algorithm for traveling salesman problem[J]. Int J of Intelligent Computing and Cybernetics, 2011, 4(3): 311-330.
- [9] Yousefikhoshbakht M, Sedighpour M. New imperialist competitive algorithm to solve the travelling salesman problem[J]. Int J of Computer Mathematics, 2013, 90(7): 1495-1550.
- [10] 冯翔, 马美怡, 虞慧群. TSP湖水能量优化算法[J]. 计算机研究与发展, 2013, 50(9): 2015-2027.
(Feng X, Ma M Y, Yu H Q. Lake-energy optimization algorithm for traveling salesman problem[J]. J of Computer Research and Development, 2013, 50(9): 2015-2027.)
- [11] Dowlatshahi M B, Nezamabadi H, Mashinchia M. A discrete gravitational search algorithm for solving combinatorial optimization problems[J]. Information Sciences, 2014, 258(10): 94-107.
- [12] 吴虎胜, 张凤鸣, 吴庐山. 一种新的群体智能算法——狼群算法[J]. 系统工程与电子技术, 2013, 35(11): 3430-3438.
(Wu H S, Zhang F M, Wu L S. New swarm intelligence algorithm — wolf pack algorithm[J]. Systems Engineering and Electronics, 2013, 35(11): 3430-3438.)
- [13] 吴虎胜, 张凤鸣, 战仁军, 等. 求解0-1背包问题的二进制狼群算法[J]. 系统工程与电子技术, 2014, 36(8): 1660-1667.
(Wu H S, Zhang F M, Zhan R J, et al. A binary wolf pack algorithm for solving 0-1 knapsack problem[J]. Systems Engineering and Electronics, 2014, 36(8): 1660-1667.)
- [14] Vlachos M, Kollios G, Gunopulos G. Discovering Similar Multidimensional Trajectories[C]. Proc of the 18th Int Conf on Data Engineering. San Jose: IEEE Press, 2002: 673-684.
- [15] 王宇平, 李英华. 求解TSP的量子遗传算法[J]. 计算机学报, 2007, 30(5): 748-755.
(Wang Y P, Li Y H. A novel quantum genetic algorithm for TSP[J]. Chinese J of Computers, 2007, 30(5): 748-755.)
- [16] 顾大权, 刘高飞, 袁珏, 等. 基于点线回路优化求解TSP的一个新算法[J]. 解放军理工大学学报: 自然科学版, 2010, 11(4): 423-427.
(Gu D Q, Liu G F, Yuan Y, et al. New algorithm for solving TSP based on dots and lines loop optimizing[J]. J of PLA University of Science and Technology: Natural Science Edition, 2010, 11(4): 423-427.)
- [17] 柳寅, 马良. 模糊人工蜂群算法的旅行商问题求解[J]. 计算机应用研究, 2013, 30(9): 2694-2696.
(Liu Y, Ma L. Fuzzy artificial bees colony algorithm for solving traveling salesman problem [J]. Application Research of Computers, 2013, 30(9): 2694-2696.)
- [18] 胡中华, 赵敏. 基于人工蜂群算法的TSP仿真[J]. 北京理工大学学报, 2009, 29(11): 978-982.
(Hu Z H, Zhao M. Simulation on traveling salesman problem(TSP) based on artificial bees colony algorithm[J]. Trans on Beijing Institute of Technology, 2009, 29(11): 978-982.)
- [19] 罗中良, 易明珠, 刘小勇. 最优化问题的蚁群混合差分进化算法研究[J]. 中山大学学报: 自然科学版, 2008, 47(3): 33-36.
(Luo Z L, Yi M Z, Liu X Y. On ant colony hybrid differential evolution for optimization problem[J]. Acta Scientiarum Naturalium Universitatis Sunyatseni: Natural Science Edition, 2008, 47(3): 33-36.)
- [20] David B. Simulated annealing for traveling salesman problem[R]. Springfield: Spring, 1997.
- [21] Bayram H. A new simulated annealing approach for Traveling salesman problem[J]. Mathematical and Computational Applications, 2013, 18(3): 313-322.
- [22] 丁海利, 王芳, 高成修. 旅行商问题的交叉粒子群优化算法[J]. 数学杂志, 2008, 28(1): 85-89.
(Ding H L, Wang F, Gao C X. Crossover particle swarm optimization for traveling salesman problem[J]. J of Mathematics, 2008, 28(1): 85-89.)
- [23] Hong K, Shi J H, Tzong W K, et al. A hybrid swarm intelligence algorithm for the traveling salesman problem[J]. Expert Systems, 2010, 27(3): 166-179.
- [24] Dong G F, Guo W W, Tichkle K. Solving the traveling salesman problem using cooperative genetic ant systems[J]. Expert Systems with Applications, 2012, 35(9): 5006-5011.
- [25] Chen S M, Chien C Y. Solving the traveling salesman problem based on the genetic simulated annealing ant colony system with particle swarm optimization techniques[J]. Expert Systems with Applications, 2011, 38(12): 14439-14450.