本文引用格式：郭文强,杜正毅.融合动态邻域搜索机制的蚁群系统算法[J].自动化与信息工程,2022,43(2):15-22.
GUO Wenqiang, DU Zhengyi. Ant colony system algorithm combining dynamic neighborhood search mechanism[J]. Automation & Information Engineering, 2022,43(2):15-22.

融合动态邻域搜索机制的蚁群系统算法*

郭文强　杜正毅
（新疆财经大学信息管理学院，新疆 乌鲁木齐 830012）

摘要：针对蚁群算法收敛速度慢，容易陷入局部最优解等问题，提出一种融合动态邻域搜索机制的蚁群系统（DNS-ACS）算法。首先，在蚁群系统算法的基础上，引入 2-opt 算子进行局部搜索，加快算法收敛速度；然后，动态调整邻域搜索范围和信息素更新规则，使其跳出局部最优，避免早熟现象；最后，利用本文提出的 DNS-ACS 算法对 16 个经典 TSP 算例进行仿真模拟，并与其他算法进行对比实验。实验结果表明，DNS-ACS 算法的求解精度明显提高，证明了该算法的有效性。

关键词：蚁群算法；旅行商问题；动态邻域搜索；信息素更新

中图分类号：TP301.6　　**文献标识码：**A　　**文章编号：**1674-2605(2022)02-0003-08

DOI：10.3969/j.issn.1674-2605.2022.02.003

0　引言

旅行商问题（travelling salesman problem, TSP）是指给定一系列城市和每两座城市之间的距离，求解访问每座城市一次并回到起始城市的最短回路，是一类标准的组合离散优化问题[1]，也是一个 NP 难问题。

TSP 有精确算法和启发式算法 2 种求解方法。其中，精确算法包括切割平面法、动态规划法、分支定界法[2]等，该类方法可有效求得 TSP 的最优解，但需要指数时间，不适用于大规模的 TSP；启发式方法包括蚁群优化算法（ant colony optimization, ACO）[3]、帝国竞争算法[4]、模拟退火法[5]、猫群算法[6]、禁忌搜索算法[7]、粒子群优化算法[8]、人工蜂群算法[9]等，该类算法能够在有限的时间内得到问题的满意解，是解决 TSP 的常用方法。

蚁群算法是具有代表性的群智能算法之一[10]，由 DORIGO Marco 于 1992 年提出，并将其应用于解决 TSP，取得较好效果。之后，各国学者不断优化该算法[11-15]，出现了最大最小蚁群（max min ant colony system, MMAS）算法[16]，蚁群系统（ant colony system, ACS）算法[17]等，这些算法优化了信息素更新规则，一定程度地解决了蚁群算法中信息素积累速度过快，搜索空间不足的问题，但仍然容易陷入局部最优。

为此，本文提出融合动态邻域搜索机制的蚁群系统（dynamic neighborhood search ant colony system, DNS-ACS）算法。首先，利用 2-opt 算子改进当前阶段的解；然后，搜索当前最优解各个节点周围邻域并记录邻域内的路径，在信息素更新阶段将信息素按照一定规则分配到记录的路径中；最后，动态调整信息素释放量以及邻域搜索范围，使算法在前期实现快速收敛，在后期有效跳出局部最优解。

1　蚁群算法

蚁群算法的灵感来源于蚂蚁觅食方式。蚂蚁在寻找食物时，在路径上遗留一种叫做信息素的化学物质。由于目标距离及挥发因素的影响，蚁群和最近食物之间的路径存在更多的信息素，从而吸引更多的蚂蚁跟随相同的路径。一旦食物耗尽，蚂蚁就会放弃这条路径，伴随着信息素的挥发，迫使蚂蚁重新开始随机寻找另一种食物。

蚁群算法随机初始化所有蚂蚁，每个蚂蚁搜索一个潜在的解决方案。在迭代过程中，蚁群算法为求解路径的每条边分配信息素，蚂蚁根据信息素的浓度，

* 基金项目：国家自然科学基金项目（61941205）；天山雪松项目（2020XS07）。

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

移动到尚未访问的节点，构建一个可行的解决方案。在蚁群算法中，蚂蚁根据公式(1)访问下一个节点。

$$P_{ij}^{k}(t)=\frac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}}{\sum_{l\in N_i^k}[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}} \quad \text{if } j\in N_i^k \tag{1}$$

式中：

$P_{ij}^{k}(t)$——状态转移概率；

$\tau_{ij}(t)$——路径 (i,j) 的信息素浓度；

$\eta_{ij}(t)$——启发函数，$\eta_{ij}(t)=1/d_{ij}$；

α、β——决定信息素和启发函数重要性的系数；

N_i^k——第k只蚂蚁当前允许访问节点的集合。

在蚁群系统算法中，蚂蚁访问节点的方式有所改变，访问规则如公式(2)所示。

$$P_{ij}^{k}(t)=\begin{cases}\arg\max\{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}\} & \text{if } q<q_0\\ S & \text{otherwise}\end{cases} \tag{2}$$

式中：

q——随机生成的数字，$q\in[0,1]$；

q_0——固定参数。

S 的访问方式按公式(1)进行。

蚁群系统算法中，蚂蚁访问下一个节点时，利用公式(3)局部更新路径信息素，这个过程称为局部信息素更新。

$$\tau_{ij}(t+1)=(1-\gamma)\tau_{ij}(t)+\gamma\tau_0 \tag{3}$$

式中：

γ——局部信息素挥发速率；

τ_0——初始信息素值。

在所有蚂蚁访问结束后，利用公式(4)更新路径信息素，这个过程称为全局信息素更新。

$$\tau_{ij}(t+1)=(1-\rho)\tau_{ij}(t)+\rho\Delta\tau_{ij}(t) \tag{4}$$

式中：

ρ——全局信息素挥发速率。

$$\Delta\tau_{ij}(t)=\begin{cases}\dfrac{1}{L_{\mathrm{gb}}(t)} & \text{if } (i,j)\in \text{best tour}\\ 0 & \text{otherwise}\end{cases} \tag{5}$$

式中：

$\Delta\tau_{ij}(t)$——蚂蚁释放在路径 (i,j) 的信息素浓度变化量；

$L_{\mathrm{gb}}(t)$——全局最佳蚂蚁走过的路径长度；

best tour——当前最优路径。

由于算法在搜索阶段开始时，各条路径的信息素分布均匀，因此在蚁群算法和蚁群系统算法中，前期具有一定的盲目性，导致收敛速度较慢；而后期，算法在路径上积累了大量的信息素，降低了搜索其他路径的机会。本文在蚁群系统算法的基础上，提出的DNS-ACS 算法在一定程度上解决了这些问题。

2 DNS-ACS 算法

本文提出的 DNS-ACS 算法主要进行如下改进：

1） 引入 2-opt 算子，避免路径交叉带来的距离损耗，实现算法快速收敛；

2） 融合邻域搜索机制，通过搜索邻域节点，扩大搜索范围，避免算法陷入局部最优；

3） 动态调整邻域搜索及信息素更新规则，为寻找最优路径提供合理导向。

2.1 2-opt 算子

2-opt 算法是局部搜索算法，通过对局部解进行优化，帮助蚁群算法避免局部极小值。本文引入 2-opt 算子求解 TSP[18]，通过将当前解所表示的路径中已有的两条边置换，产生一条新路径，若新路径更短，则保留；重复这个过程，直到没有发现更短的路径为止。2-opt 算子的优化过程如图 1 所示。

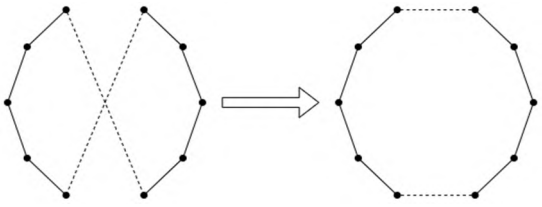


图 1 2-opt 算子优化过程示意图

由图 1 可以看出，2-opt 算子能够将虚线部分的路径优化，使优化后的路径更短，有效提升问题解的质量。

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

2.2 邻域搜索机制

蚂蚁在寻找下一个目标节点时，符合最优解的目标往往在蚂蚁周围的相邻节点产生。因此，在 DNS-ACS 算法中，蚂蚁经过每个节点时，标记周围距离最近的相邻节点，并记录当前节点 M 与标记点之间的路径，标记个数为 NN。蚂蚁在经过某个节点时，建立邻域，寻找标记点及与当前节点路径的过程如图 2 所示。

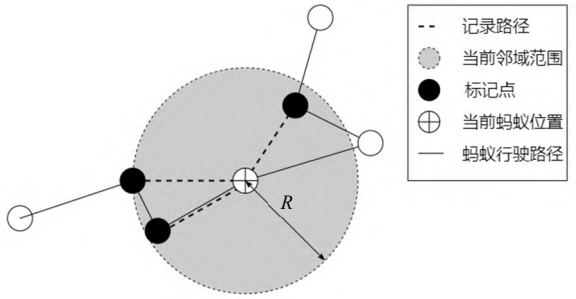


图 2 邻域搜索机制示意图

由图 2 可以看出，蚂蚁经过每个节点时，会建立以该节点为中心，半径为 R 的邻域范围。R 的计算公式为

$$R=\max\{dist(M,P_i)\},\quad i\in[1,NN] \tag{6}$$

式中：

M——当前节点；

P_i——当前节点对应的各个标记点。

2.3 信息素动态更新

在蚁群算法中，每只蚂蚁经过路径时都释放信息素，导致信息素积累速度过快。随着迭代次数不断增加，蚂蚁容易集中在同一条路径，导致收敛过快，结果误差偏大。蚁群系统算法通过局部信息素更新和全局信息素更新，避免了路径间信息素差异过大。但由于信息素更新方式是静态的，仍难以跳出局部最优解。

本文通过动态调整邻域的信息素，使其在加快算法收敛速度的同时扩大搜索范围，避免早熟现象。该过程分为 2 个步骤。

1） 对现有的最优路径(i,j)按公式(7)进行信息素补偿，以加快算法收敛速度。

$$\tau_{ij}(t+1)=\tau_{ij}(t)+\Delta\tau_{ij}^{*}(t)\quad (i,j)\in L_{\text{best}} \tag{7}$$

其中 $\Delta\tau_{ij}^{*}(t)$ 按公式(8)分 2 种情况进行：

$$\Delta\tau_{ij}^{*}(t)=\begin{cases}\dfrac{1}{U(n\cdot R)} & D_{ij}<R\\[2ex] \dfrac{1}{U(n\cdot D_{ij})} & \text{otherwise}\end{cases} \tag{8}$$

式中：

$\Delta\tau_{ij}^{*}(t)$——信息素补偿量；

U——信息素补偿系数；

D_{ij}——路径(i,j)长度。

即若节点 j 处于其当前邻域范围内，则信息素的补充值为一个与领域范围半径 R 相关的固定值；若节点 j 处于当前邻域范围外，则信息素的补充值随两点距离的增大而减小。

2） 为平衡可能存在的更优解路径与现有最优路径之间的信息素差距，向当前最优路径节点邻域内的其他节点释放信息素，增加蚂蚁前往这些节点的概率，扩大搜索范围。

首先，利用公式(9)对某一节点记录的各条路径间的距离进行标准化处理；

$$\phi_{[(t,i),(t,j)]}=\frac{d_{[(t,i),(t,j)]}}{\sum_{l=1}^{NN}d_{[(t,i),(t,l)]}}\quad j\in[1,NN] \tag{9}$$

然后，利用公式(10)对记录的路径进行信息素更新。

$$\tau_{ip}(t+1)=\tau_{ip}(t)+\left(\frac{\tau_{ij}L_{\text{best}}}{I\cdot L_{\text{ave}}(t)}\right)\cdot\left(\frac{1}{1+\sqrt{\phi_{ip}}}\right) \tag{10}$$

式中：

I——迭代次数；

L_{best}——当前迭代求得的局部最优解；

L_{ave}——迭代求得的平均解。

随着迭代次数增加，信息素聚集，结果趋于收敛，平均解减小；而平均解减小会释放更多的信息素，以加大下次选取节点时周围路径的信息素权重，增大蚂蚁选择该路径的概率，有利于跳出局部最优。如果寻得更优的 L_{best}，则适当降低信息素，使算法趋于收敛。

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

2.4 动态邻域搜索机制

在邻域搜索时，若每次迭代的结果较上一次有所提升，则表明算法在当前环境下能够找到更优解的可能性较大，应适当缩小搜索范围，集中搜索可能出现的更优解；若长时间最优解没有更新，则表明算法陷入停滞，应扩大搜索范围，增加解的多样性，在更远的邻域内使现有解得到更新。本文对标记数 *NN* 按公式(11)进行调整。

$$NN=\begin{cases} NN-1 & L_{\text{best}}^{I}<L_{\text{best}}^{I-1} \\ NN+1 & L_{\text{best}}^{I}=L_{\text{best}}^{I-\frac{I_{\max}}{5}} \end{cases} \tag{11}$$

式中：

L_{best}^{I}——第I代寻优过程中的最优解；

$I_{\max}$——最大迭代次数。

为避免邻域范围过大或过小，干扰蚂蚁寻优过程，设定一个阈值，如公式(12)所示。

$$NN\in[2,n/n_0] \tag{12}$$

式中：

n——蚂蚁访问节点的个数；

n_0——固定参数。

2.5 算法流程

DNS-ACS 算法流程图如图 3 所示。

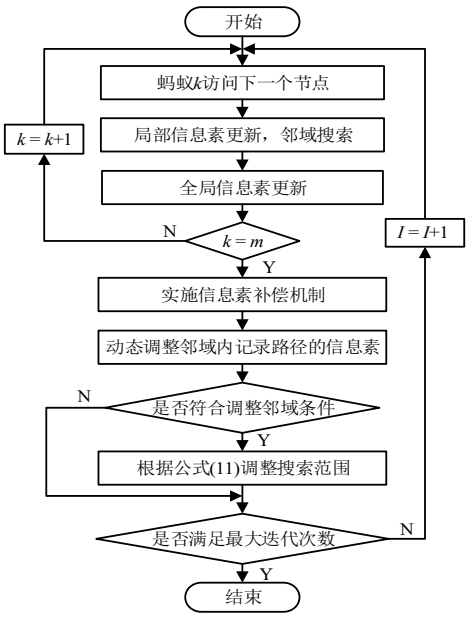


图 3　算法流程图

3　实验仿真

3.1　参数设置

本实验采用 TSPLIB 数据库中的 TSP 数据集，测试算法性能。首先，在数据集中选取 16 个算例，比较 ACS、ACS+2-opt、本文 DNS-ACS 算法；其次，利用 8 个数据集比较本文 DNS-ACS 算法与 5 种文献[5,19-22]算法性能。参数设置如表 1 所示。

表 1　参数设置

参数符号	值	参数符号	值
$I_{\max}$	100	ρ	0.3
U	10	n_0	30
m	100	γ	0.1
a	1	q_0	0.5
b	4		

通过 Matlab 2018b 实现 DNS-ACS 算法，并分别运行 TSP 数据集的 16 个算例，比较其平均解，误差率（*Er*）以及每次运行的最优解。

3.2　与传统算法比较

为验证 DNS-ACS 算法能有效优化 TSP，本文利用 ACS、ACS+2-opt、DNS-ACS 三种算法分别对 TSP 算例库中的 16 个算例进行实验。为保证参数的一致性，三种算法的参数都按表 1 设置。每种算法在每个算例中运行 10 次，取最优解平均值，并利用公式(13)计算误差率，结果如表 2 所示。

$$Er=\frac{C_{\text{Best}}-C_{\text{BKS}}}{C_{\text{BKS}}}\% \tag{13}$$

式中：

Er——误差率；

C_{Best}——算法求解算例时所得的最优解；

C_{BKS}——当前算例已知的最优解。

由表 2 可知：ACS 算法仅找到 oliver30 算例的最优解，其他算例都有不同程度的误差，其中 lin318 算例的误差率达到 7.93%；ACS+2-opt 算法找到 9 个算例的最优解，其中误差率高于 0.5%的算例有 3 个；DNS-ACS 算法找到 12 个算例的最优解，其他算例的误差率在 0.5%以内，寻优能力明显优于另外 2 种算

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

法。此外，DNS-ACS 算法在每个算例中的最优值，都　　是 3 种算法中最小的，证明了该算法的优化是有效的。

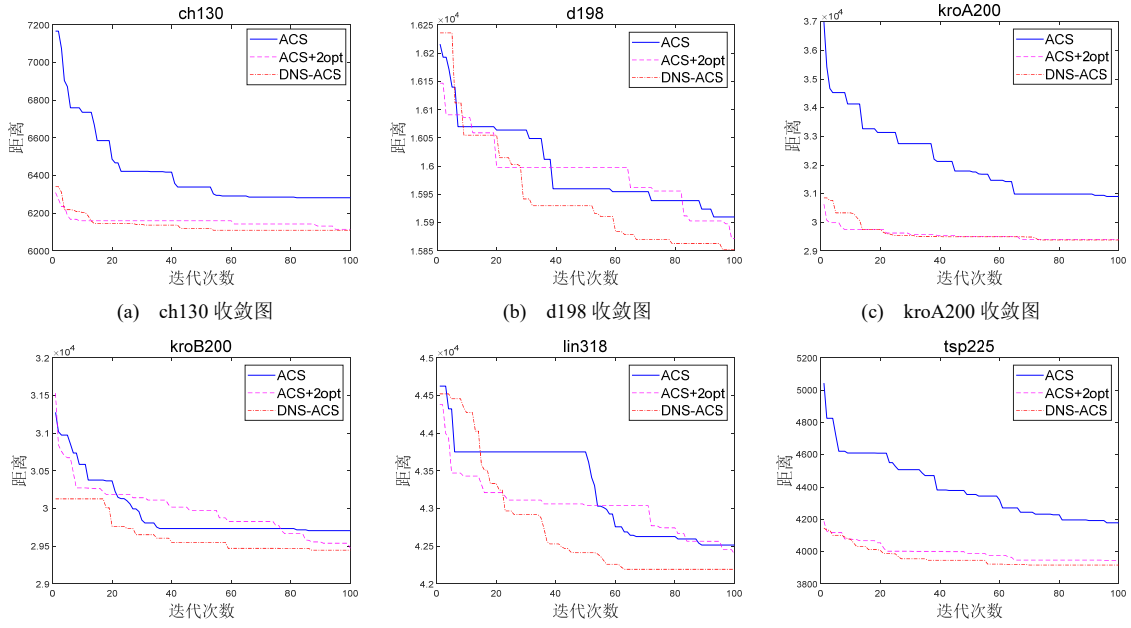
表 2　DNS-ACS 算法与 ACS 算法、ACS+2-opt 算法对比结果

TSP 算例	已知最优解	ACS 算法			ACS+2-opt 算法			本文 DNS-ACS 算法		
		最优解	*Er*/%	平均解	最优解	*Er*/%	平均解	最优解	*Er*/%	平均解
eil51	426	429	0.70	436	**426**	0.00	427	**426**	0.00	**426**
eil76	538	544	1.12	554	**538**	0.00	539	**538**	0.00	**538**
eil101	629	653	3.82	673	**629**	0.00	633	**629**	0.00	631
ch130	6 110	6 282	2.82	6 392	6 115	0.08	6 143	**6 110**	0.00	6 135
d198	15 780	16 652	5.53	17 172	15 872	0.58	15 947	15 852	0.46	15 894
kroA100	21 282	21 987	3.31	22 231	**21 282**	0.00	21 284	**21 282**	0.00	**21 282**
kroB100	22 141	22 478	1.52	22 632	**22 141**	0.00	22 160	**22 141**	0.00	22 147
kroA150	26 524	27 489	3.64	28 148	26 550	0.10	26 659	**26 524**	0.00	26 598
kroB150	26 130	27 401	4.86	27 889	**26 130**	0.00	26 218	**26 130**	0.00	26 149
kroA200	29 368	30 900	5.22	31 560	29 397	0.10	29 495	29 383	0.05	29 443
kroB200	29 437	30 877	4.89	31 886	29 473	0.12	29 644	29 447	0.03	29 570
lin318	42 029	45 364	7.93	46 478	42 391	0.86	42 708	42 193	0.39	42 617
oliver30	420	**420**	0.00	421	**420**	0.00	**420**	**420**	0.00	**420**
rat99	1 211	1 226	1.24	1257	**1 211**	0.00	1 212	**1 211**	0.00	1 212
st70	675	690	2.22	703	**675**	0.00	**675**	**675**	0.00	**675**
tsp225	3 916	4 178	6.69	4 296	3 945	0.74	3 975	**3 916**	0.00	3 931

注：最优解和平均解的结果四舍五入取整，标黑表示达到已知最优解。

本文选择 6 个规模较大的算例，通过图像的形式　　进行对比，如图 4 所示。

对 ACS、ACS+2-opt、DNS-ACS 三种算法的收敛曲线



(a)　ch130 收敛图　　(b)　d198 收敛图　　(c)　kroA200 收敛图

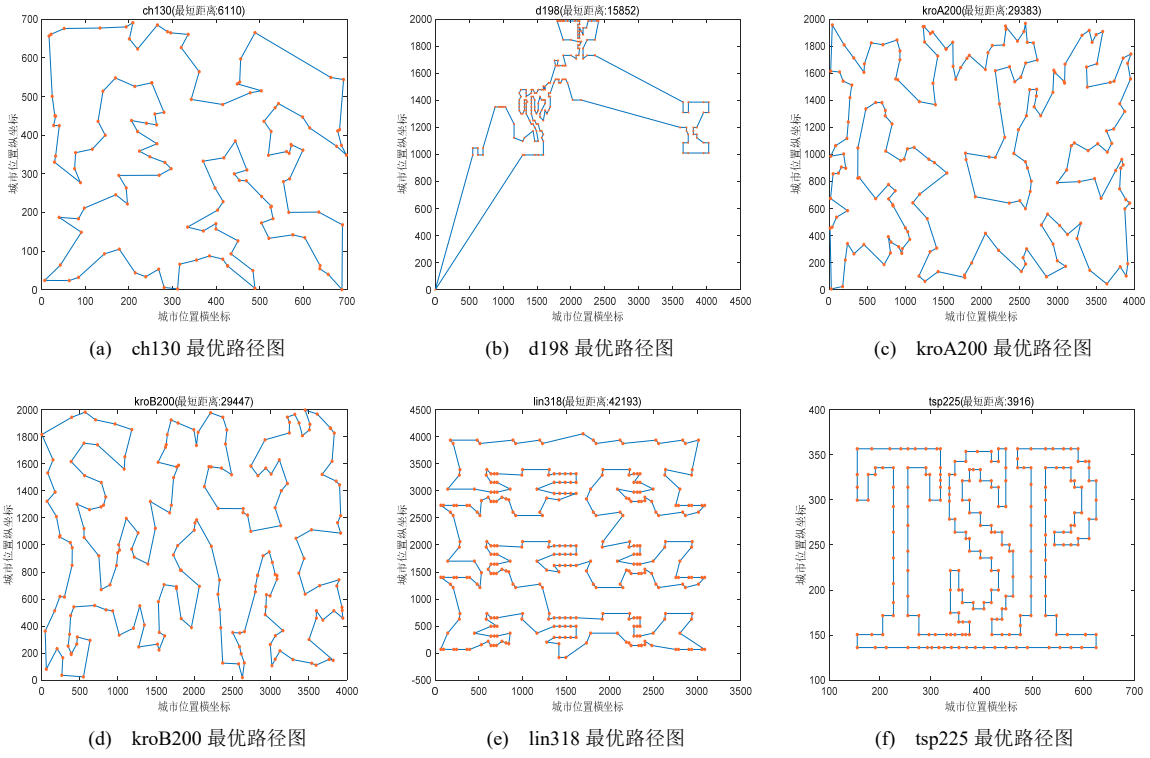
(d)　kroB200 收敛图　　(e)　lin318 收敛图　　(f)　tsp225 收敛图

图 4　ACS、ACS+2-opt、DNS-ACS 三种算法的收敛曲线对比图

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

由图 4 可知，DNS-ACS 算法的收敛速度较快，求解精度最好，表明该算法对求解 TSP 问题有明显效果。

为对仿真结果进行进一步验证，采用 DNS-ACS 算法求解的最优路径如图 5 所示。



(a) ch130 最优路径图　(b) d198 最优路径图　(c) kroA200 最优路径图

(d) kroB200 最优路径图　(e) lin318 最优路径图　(f) tsp225 最优路径图

图 5　仿真实验结果

由图 5 可以看出：采用 DNS-ACS 算法求解的最优路径是一个闭环，其间没有形成交叉点。

3.3　与其他文献算法比较

为进一步证明 DNS-ACS 算法的有效性，本文选取文献[5,19-22]的算法在这 8 个算例中进行比对实验，结果如表 3 所示。

表 3　DNS-ACS 算法与其他文献算法比对结果

算例	文献[19]		文献[20]		文献[21]		文献[22]		文献[5]		DNS-ACS	
	最优解	平均解	最优解	平均解	最优解	平均解	最优解	平均解	最优解	平均解	最优解	平均解
eil51	**426**	428	**426**	430	427	431	**426**	427	**426**	430	**426**	**426**
st70	676	680	676	682	—	—	—	—	**675**	681	**675**	**675**
eil76	**538**	547	**538**	541	**538**	546	**538**	540	**538**	541	**538**	**538**
rat99	**1 211**	1 217	**1 211**	1 213	1 216	1222	—	—	**1 211**	1 228	**1 211**	**1 212**
kroA100	21 292	21 297	21 308	21 445	—	—	**21 282**	21 345	**21 282**	21 346	**21 282**	**21 282**
eil101	—	—	631	639	639	662	—	—	**629**	641	**629**	**631**
kroB200	—	—	29 581	29 781	—	—	29 390	29 768	29 401	30 144	**29 383**	**29 443**
tsp225	—	—	—	—	—	—	3 931	3 942	3 945	4 052	**3 916**	**3 931**

注：最优解和平均解的结果四舍五入取整，“—”表示在文献中没有相应数据，标黑表示算法中的最优解。

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

由表 3 可以看出：在这 8 个算例中，DNS-ACS 算法解的精度较其他算法有明显提高，从而验证该算法具有较好的优化性能。

4 结语

针对蚁群算法的不足，本文融入动态邻域搜索机制，结合 2-opt 算子，并通过改进信息素更新机制，在加快算法收敛速度的同时，扩大搜索范围，从而使算法更好地跳出局部最优解，实现算法优化。通过对 TSPLIB 数据库中的 16 个算例的比对，证明该算法的有效性。但是对于较大规模的算例，所求得解的精度仍有上升的空间。日后的研究中，将针对这一问题继续完善对算法的优化。

参考文献

[1] DANTZIG G B, FULKERSON D R, JOHNSON S M. On a linear-programming, combinatorial approach to the traveling-salesman problem[J]. Operations Research, 1959,7(1):58-66.

[2] GOUVEIA L, LEITNER M, RUTHMAIR M. Extended formulations and branch-and-cut algorithms for the black-and-white traveling salesman problem[J]. European Journal of Operational Research, 2017:908-928.

[3] WEI Gao. New ant colony optimization algorithm for the traveling salesman problem[J]. International Journal of Computational Intelligence Systems,2020,13(1): 44-55.

[4] ARDALAN Z, KARIMI S, POURSABZI O, et al. A novel imperialist competitive algorithm for generalized traveling salesman problems[J]. Applied Soft Computing, 2015,26:546- 555.

[5] 陈科胜,鲜思东,郭鹏.求解旅行商问题的自适应升温模拟退火算法[J].控制理论与应用,2021,38(2):245-254.

[6] 杨进,郑允,马良.改进的猫群算法求解 TSP[J].计算机应用研究,2017,34(12):3607-3610.

[7] EZUGWU E S, ADEWUMI A O, FRNCU M E. Simulated annealing based symbiotic organisms search optimization algorithm for traveling salesman problem[J]. Expert Systems with Applications, 2017,77:189-210.

[8] 杨云亭,王鹏.求解旅行商问题的多尺度量子自由粒子优化算法[J].计算机应用,2020,40(5):1278-1283.

[9] CHOONG S S, WONG L P, LIM C P. An artificial bee colony algorithm with a Modified Choice Function for the traveling salesman problem [C]// IEEE International Conference on System, Man, and Cybernetics (SMC) 2017. IEEE, 2017.

[10] DORIGO M, MANIEZZO V, COLORNI A. Ant system: optimization by a colony of cooperating agents[J]. IEEE transactions on systems, man and cybernetics. Part B, 1996,26(1): 29-41.

[11] ZHANG Zhaojun, XU Zhaoxiong, LUAN Shengyang, et al. Opposition-based ant colony optimization algorithm for the traveling salesman problem[J]. Mathematics, 2020,8(10):1650.

[12] EBADINEZHAD S. DEACO: adopting dynamic evaporation strategy to enhance ACO algorithm for the traveling salesman problem[J]. Engineering Applications of Artificial Intelligence, 2020,92:103649.

[13] FADL D, KHALIL EL H, HASSAN M, et al. Dynamic flying ant colony optimization (DFACO) for solving the traveling salesman problem[J]. Sensors (Basel, Switzerland),2019,19(8): 1837.

[14] JIANG C, WAN Z, PENG Z. A new efficient hybrid algorithm for large scale multiple traveling salesman problems[J]. Expert Systems with Applications, 2019,139:112867.

[15] 张立毅,肖超,费腾.基于细菌觅食的改进蚁群算法[J].计算机工程与科学,2018,40(10):1882-1889.

[16] STUTZLE T, HOOS H H. MAX-MIN ant system[J]. Future Generation Computer Systems, 2000, 16(8): 889-914.

[17] DORIGO M, GAMBARDELLA L M. Ant colony system: a cooperative learning approach to the traveling salesman problem[J]. IEEE Transactions on Evolutionary Computation, 1997,1(1):53-66.

[18] 李俊,童钊,王政.一种并行 ACS-2-opt 算法处理 TSP 问题的方法[J].计算机科学,2018,45(S2):138-142.

[19] 陈思远,林丕源,黄沛杰.指针网络改进遗传算法求解旅行商问题[J].计算机工程与应用,2020,56(19):231-236.

[20] 乔东平,裴杰,李浩,等.改进蚁群算法求解 TSP 问题研究[J].机械设计与制造,2019(10):144-149.

[21] 陈颖杰,高茂庭.基于信息素初始分配和动态更新的蚁群算法[J].计算机工程与应用,2022,58(2):95-101.

[22] 孟静雯,游晓明,刘升.结合协同机制与动态调控策略的双蚁群算法[J].计算机科学与探索,2021,15(11):2206-2221.

Ant Colony System Algorithm Combining Dynamic Neighborhood Search Mechanism

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

GUO Wenqiang　DU Zhengyi

(School of Information Management　Xinjiang University of Finance and Economics, Urumqi 830012, China)

Abstract: Aiming at the problems of slow convergence speed and easy to fall into local optimal solution of ant colony algorithm, an ant colony system (DNS-ACS) algorithm combining dynamic neighborhood search mechanism is proposed. Firstly, based on the ant colony system algorithm, the 2-opt operator is introduced for local search to speed up the convergence speed of the algorithm; Then, dynamically adjust the neighborhood search range and pheromone update rules to make it jump out of the local optimum and avoid premature phenomenon; Finally, the DNS-ACS algorithm proposed in this paper is used to simulate 16 classical TSP examples, and compared with other algorithms. Experimental results show that the accuracy of DNS-ACS algorithm is significantly improved, which proves the effectiveness of the algorithm.

Keywords: ant colony algorithm; traveling salesman problem; dynamic neighbors search; pheromone update

作者简介：

郭文强，男，1975 年生，博士，教授，博士生导师，主要研究方向：人工智能、信息安全。E-mail: gwq@xjufe.edu.cn

杜正毅，男，1996 年生，硕士研究生，主要研究方向：智能算法。E-mail:1247397930@qq.com

（上接第 7 页）

Study on Parameters and Properties of Laser Cladding 60%WC-Ni Coating

ZHANG Li[1, 2]　BI Guijun[1, 2]　CAO Lichao[2]　CHANG Yunlong[1, 3]

(1. Guangdong, CAS Dofortune Laser Technology Co., Ltd. Foshan 200240, China
2. Institute of Intelligent Manufacturing, GDAS, Guangzhou 510070, China
3. School of Materials Science and Engineering, Shenyang University of Technology, Shenyang 110000, China)

Abstract: WC-Ni matrix composite is one of the commonly used laser cladding materials, it is able to effectively improve the wear resistance of the material surface. However, as WC is a hard and brittle material, it is easy to decompose, dissolve and oxidize when heated, leading to the limited volume fraction of WC in the cladding layer, and the coating is prone to crack. Therefore, the current laser cladding research on this material mainly focuses on the low WC content (mass fraction<50%). In order to further explore the laser cladding properties of high content WC nickel-based alloy (mass fraction>50%), 60%WC-Ni powder was used as laser cladding material, and CCS-B steel plate was used as the base material in this work. The effects of laser power, powder feeding rate and cladding speed on the width, height and dilution rate of the cladding layer were studied respectively. Finally, the appropriate combination of cladding parameters was determined, and the cladding layer was formed on the surface of the substrate. The hardness, friction and wear tests of substrate and cladding layer were carried out respectively, results showed that the average hardness of the cladding layer is 81.44HRC, 5.45 times higher than that of the matrix, the abrasion loss and wear coefficient were reduced by 93.6% and 12.37% in the same time. The hardness and wear resistance of the cladding layer are significantly higher than that of the matrix ascension.

Keywords: laser cladding; process parameters; hardness; wear resistance

作者简介：

张理，男，1990 年生，博士研究生、助理研究员，主要研究方向：激光增材制造。E-mail: l.zhang@giim.ac.cn

毕贵军，男，1971 年生，博导，研究员，主要研究方向：激光增材制造。

曹立超，男，1991 年生，硕士，工程师，主要研究方向：爬壁机器人、增材制造。

常云龙，男，1963 年生，博导，教授，主要研究方向：焊接及增材制造。

(C)1994-2022 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net