文章编号: 1001-0920(2017)03-0552-05 DOI: 10.13195/j.kzyjc.2016.0005

一种动态搜索策略的蚁群算法及其在机器人路径规划中的应用

游晓明[1†], 刘 升[2], 吕金秋[1]

(1. 上海工程技术大学 电子电气工程学院，上海 201620；2. 上海工程技术大学 管理学院，上海 201620)

摘 要: 采用蚁群算法求解复杂环境下移动机器人路径规划问题时,会出现运算时间过长、求解精度不高等问题,对此,定义一种新的动态搜索诱导算子以改进蚁群算法性能. 重点设计了动态搜索模型,即:在进化初期设定较大阈值以增加种群的多样性;而伴随进化过程,利用衰减模型动态调整为较小阈值以加快收敛速度. TSP测试实验结果表明,该改进蚁群算法不仅能加快收敛速度,而且有效提高了优化解的质量. 复杂环境中机器人路径规划问题的求解验证了所提出算法的实际应用效果.

关键词: 蚁群系统；动态搜索诱导算子；移动机器人；路径规划；复杂环境

中图分类号: TP273 **文献标志码:** A

Ant colony algorithm based on dynamic search strategy and its application on path planning of robot

YOU Xiao-ming[1†], *LIU Sheng*[2], *LV Jin-qiu*[1]

(1. College of Electronic and Electrical Engineering，Shanghai University of Engineering Science，Shanghai 201620，China；2. School of Management, Shanghai University of Engineering Science，Shanghai 201620，China)

Abstract: To overcome difficulties of the traditional ant colony optimization, a novel ant colony system based on dynamic search(DSACS) strategy for path planning problem of mobile robot is proposed. Therefore, a dynamic search model is designed. in the prophase, a bigger parameter is used to increase the diversity of the population; in the anaphase, a smaller parameter is adjusted through the attenuation model to accelerate convergence. Experimental results of TSP benchmark instances show that the improved ant colony algorithm can not only accelerate the convergence, but also improve the quality of the optimal solution. Simulation results of path planning problems under the complex environment verify the cutual effect of the DSACS strategy.

Keywords: ant colony system；dynamic search bias operator；mobile robot；path planning；complex environment

0 引 言

意大利学者Dorigo等[1]受生物进化机理的启发,提出了蚁群算法(ACO),已成功地应用于许多实际工程优化问题,包括二次分配问题、机器人路径规划[2]、网络路由选择问题[3]等. 蚁群算法具有天然的并行性、较强的鲁棒性,易于与其他优化方法结合[4]. 但当转移概率过大时,虽有较快的收敛速度,也会导致早熟收敛问题. 特别是正反馈原理所引起的自催化现象,虽然能够强化性能较好解,但也容易出现停滞现象等问题.

面对大规模复杂问题,蚁群算法存在收敛速度慢、容易出现停滞等问题,学者们纷纷提出了各种改进策略[5-10]. 如Dorigo等[1]提出了蚁群系统(ACS)算法,通过对每一代中最好个体所走过的路径上的信息素进行更新,以加快收敛速度,但由于强化了最优信息的反馈,易导致停滞现象. Stutzle等[5]提出的最大最小蚂蚁系统(MMAS),通过对路径上信息素的上下限进行限定,在一定程度上避免了停滞现象,但当解的分布较分散时,收敛速度也会变慢. 大量文献研究均表明[6-10],混合蚁群算法能有效平衡加快收敛速度和避免局域最优解,因此,更易于获得优化解. 文献[6]中采用PSO的局域和全局搜索机制来改进ACO

收稿日期: 2016-01-04；**修回日期:** 2016-07-25.

基金项目: 国家自然科学基金项目(61075115, 61403249, 61673258).

作者简介: 游晓明(1963−), 女, 教授, 博士, 从事智能信息处理、机器人技术等研究；吕金秋(1991−), 男, 硕士, 从事智能算法优化、嵌入式系统的研究.

†通讯作者. E-mail: yxm6301@163.com

的信息素更新策略. 文献[7]首先利用PSO和GA的随机、快速和全局性获得次优解序列,并调整ACO的初始信息素分布;然后利用高质量解完成最优解. 文献[8]则采用两阶段混合法:第1阶段利用快速反向梯度法获取可能的优化解;第2阶段采用ACO算法有效提高优化解的质量. 上述文献算法的实验结果均表明了各自的优越性.

蚁群算法按照搜索方向进行搜索,一旦信息素引导错误的搜索方向,搜索过程将会陷入局域最优. 目前,对算法的改进工作主要是提高种群的多样性,较少有对搜索方向进行控制[10],尤其是如何利用动态搜索策略有效控制搜索方向的方法甚少. 鉴于此,本文提出一种混合动态搜索诱导算子和3-opt算子的蚁群系统算法(DSACS). 其中:动态搜索诱导算子可以有效控制算法的搜索方向并提高收敛速度,而3-opt算子[11]的局域优化能力可以进一步提高优化解的质量,从而有效地避免早熟收敛. 文中探讨了动态搜索模型的性能,分别给出模拟退火搜索算子和刻度衰减搜索算子,并通过实验分析验证模拟退火搜索算子性能更加优越.最后分别通过求解TSP问题和机器人路径规划问题验证了本文算法的优良性能和效果.

1　相关研究

1.1　蚁群算法及相关工作

蚁群算法的数学模型描述[2]:具有组合优化性质的最小化问题的构造图为$G=\langle V,E\rangle,|V|=N,N$是图上结点数,$E=\{(\nu_i,\nu_j):\nu_i,\nu_j\in V\}$,图$G$是$N$阶无向完全图. 全状态集合是$X$,其元素由$V$中所有可能的序列组成,它是有限集. 即图$G$上每条通路对应于集合$X$中的一个元素,可行解集$\tilde{X}\subseteq X$由满足约束条件的通路组成,其中全局最优解集$S^*\subseteq\tilde{X}$. 通路的目标函数(代价)用$f$表示. 首先在图$G$上随机分布蚁群,根据具体问题选择$s$个起始结点,设置每只蚂蚁的禁忌表$\mathrm{Tabu}=\varnothing$,令$n=0$,对于$\forall e_{ij}\in E$,$\tau_{ij}(0)=\tau_0\geqslant\tau_{\min}(0)$. 第$n$次迭代过程中,蚂蚁个体$A_k$按照下式选择一个后继结点[13]:

$$P({}_{(n-1)}e_{i_kj_k})=P(X_k^{(n)}=\nu_{j_k}\mid X_k^{(n-1)}=\nu_{i_k})=$$

$$P_{ij}^k=\begin{cases}\dfrac{{\tau_{ij}^{\alpha}}\cdot\eta_{ij}^{\beta}}{\displaystyle\sum_{j\in N_i{}^k}{\tau_{i,j}}^{\alpha}\cdot{\eta_{i,j}}^{\beta}},\ j\in N_i{}^k;\\ 0,\ \text{else.}\end{cases}\tag{1}$$

其中:随机变量$X_k^{(n)}$表示蚂蚁个体A_k在第n次搜索周期结束时所处状态,第n次迭代后蚂蚁A_k发现的通路$\Gamma_k(n)={}_{(0)}e_{i_kj_k}{}_{(1)}e_{l_km_k}\cdots{}_{(n-1)}e_{h_k\mu_k},1\leqslant k\leqslant s$,表示蚂蚁个体$A_k$当前在构造图$G=\langle V,E\rangle$上所走过的结点序列,它到达$\nu_{\mu_k}$之前已经走过的结点集合记作$\mathrm{Tabu}_{\mu_k}$,$\tilde{E}$是可行解集$\tilde{X}$对应的弧段,本次迭代产生的代价最小通路记作$\Gamma^\star(n)$;$\tau_{i,j}$表示城市$i$到城市$j$路径上的信息素;$\eta_{i,j}$表示城市$i$到城市$j$之间的启发式因子,一般取$1/d_{i,j}$,即城市$i$到城市$j$间距离的倒数;$N_i^k$记忆蚂蚁$k$还没有访问过的城市.

1.2　信息素更新过程

蚂蚁个体在其所经路径上释放信息素,并根据这些区域上分布的信息素量确定该区域对其吸引力. 当全体蚂蚁都完成上述解的构造时,所有路径上的信息素以上一周期残留信息量的ρ倍挥发掉,只有最优路径$\Gamma^\star(n)$上的信息素得到补偿. 这里采用GBAS/tdlb信息素更新策略[14],有

$$\tau_{i,j}=\begin{cases}\max\left((1-\rho)\tau_{i,j}(n-1)+\dfrac{\rho}{f(\Gamma^\star(n))},\tau_{\min}(n)\right),\\ \qquad\text{if }{}_{(n-1)}e_{i,j}\in\Gamma^\star(n);\\ \max((1-\rho)\tau_{ij}(n-1),\tau_{\min}(n)),\ \text{otherwise.}\end{cases}\tag{2}$$

由式(2)可知,未被选择的路径在下一搜索周期信息素更新中,其上的信息素会因挥发而越来越少.

1.3　k-opt算子

k-opt局部搜索算法也称k交换算法. 设T是一条初始路径,该算法总是试图找到2个集合:$X=\{x_1,x_2,\cdots,x_k\},X\in T$和$Y=\{y_1,y_2,\cdots,y_k\},Y\notin T$,如果在一定要求下用$Y$集合代替$X$集合,使得新的费用代价变小,则这$k$条边的交换就被称为$k$-opt交换. 当$k=3$时,称为3-opt算法[11].

2　改进蚁群算法求解旅行商问题

本文提出一种基于动态搜索诱导算子的蚁群算法(DSACS),其中动态搜索诱导算子可有效控制算法的搜索方向并提高收敛速度,而3-opt算子进一步提高了优化解质量,因此本文算法(DSACS)具有更高效的性能.

2.1　动态搜索诱导算子

本文重点设计了动态搜索诱导算子,通过该操作算子,可以获得可行解,即:式(3)中q_{ki}^0矩阵的任意行和列有且仅有一个元素为1,则q_{ki}^0为一个可行解. 具体操作如下:

1) $M=\{1,2,\cdots,n\}$,$\mathrm{Tabu}=\varnothing$,$a$和$b$分别为矩阵的行值和列值,$a$取初值为1,$\mathrm{Tabu}=\mathrm{Tabu}\bigcup\{a\}$;$l$为迭代次数,初值$l_0=1$;$p\in[0,1]$是设定的阈值.

2)对于q_{ki}^0的a行,生成随机数γ. 如果$\gamma<p$,则按轮盘赌选择列$b\in M$,且$b\notin\mathrm{Tabu}$,令$q_{ki}^0(a,b)=1$;否则,在第a行上取式(1)中概率最大的列b,即$b=$

$\mathrm{argmax}(P_{a,j}^k)$,且$b \notin$ Tabu,令$q_{ki}^0(a,b)=1$. 取该行其余列的值为0.

3) $a=b$, $\mathrm{Tabu}=\mathrm{Tabu}\bigcup\{b\}$, $l=l+1$, $p=g(l)$, 重复2)、3)过程,直到q_{ki}^0的任意行和列有且仅有一个元素为1, $p=g(l)$是与迭代次数有关的动态阈值,本文分别考虑模拟退火函数$p=p/\log_2(1+l)$和刻度衰减模型$p=p(1-l/\mathrm{NC})$.

例如:当$n=4$时,q_{ki}^0可以为

$$\begin{bmatrix}0&0&1&0\\0&0&0&1\\0&1&0&0\\1&0&0&0\end{bmatrix},$$

表示可行路径为1-3-2-4-1.

2.2 动态搜索模型

本文提出动态搜索策略,阈值$p\in[0,1]$作为诱导因子控制收敛速度. p值越大,按照轮盘赌随机性选择概率就越大,种群多样性会越好;反之,p值越小,轮盘赌随机性选择概率就越小,而按式(1)中概率最大选择就越大,收敛速度会越快. 因此,本文采用动态搜索诱导算子,即在进化初期设定较大p值来增加种群的多样性,而在进化后期调整为较小阈值以加快收敛速度. 据此,选择随迭代次数增长而递减的衰减模型作为调整阈值p的计算模型,这里分别考虑模拟退火函数$p=p/\log_2(1+l)$和刻度衰减模型$p=p(1-l/\mathrm{NC})$来调整阈值的变化(l为循环迭代次数,NC为最大迭代次数). 由于实验研究表明模拟退火函数$p=p/\log_2(1+l)$更具优越性,本文算法采用模拟退火函数.

2.3 改进蚁群算法

DSACS算法的具体步骤如下.

1)初始化种群. 将种群进行初始化,有

$$Q_k(0)=\{q_{k1}^0,q_{k2}^0,\cdots,q_{ks}^0\},$$

$$q_{ki}^0=\begin{bmatrix}q_{ki}^0(1,1)&q_{ki}^0(1,2)&\cdots&q_{ki}^0(1,n)\\q_{ki}^0(2,1)&q_{ki}^0(2,2)&\cdots&q_{ki}^0(2,n)\\\vdots&\vdots&\ddots&\vdots\\q_{ki}^0(n,1)&q_{ki}^0(n,2)&\cdots&q_{ki}^0(n,n)\end{bmatrix},$$
$$k=1,2,\cdots,m. \tag{3}$$

式(3)为种群中的第i个个体,用$n\times n$的矩阵表示,n为城市个数.

2)循环迭代.

{

i)更新信息素[14];

ii)动态搜索诱导算子以获得可行解集;

iii)应用3-opt局部寻优算法提升可行解集中的最优和次优路径;

}

直至达到规定迭代次数或演化目标,结束循环.

3)算法结束.

3 实验研究与仿真结果

本文采用TSPLIB测试实例和机器人路径规划问题求解测试算法性能,DSACS算法用C#实现,测试结果在CPU为Intel Celeron 1.6 G的PC机上运行得到. CH144TSP测试实例运行10次的最好结果为30 334.934 585,且10次运行结果均值小于30 400. KORB150测试实例运行10次的最好结果为26 127.934 585,该结果比标准测试集中的最优结果26 130更加优化. 测试结果表明,动态搜索诱导算子可有效控制算法的搜索方向并提高收敛速度,且3-opt局部搜索策略可进一步提高优化解的质量,因此,本文算法获得了更加优化的结果.

3.1 算法性能分析

3.1.1 动态搜索模型性能

分别采用模拟退火函数$p=p/\log_2(1+l)$和刻度衰减模型$p=p(1-l/\mathrm{NC})$来调整阈值的变化

表1 ACS加入动态搜索算子的优化结果

TSP问题	测试集最优解	动态搜索算子	算法最优解	误差/%	路径长度			迭代次数		
					最大值	均值	标准差	最小值	最大值	均值
EIL51	426	ACS+模拟退火	428.81	0.65	467.355	429.38	21.671	160	400	256
		ACS+刻度衰减	428.98	0.7	469.673	430.21	24.178	200	480	368
		ACS	428.98	0.7	471.341	431.15	27.135	250	600	420
KORB150	26 130	ACS+模拟退火	26 127.93	—	26 362	26 143	122.54	1 500	1 800	1 600
		ACS+刻度衰减	26 130	0	26 535	26 175	202.34	2 000	2 500	2 200
		ACS	26 335	0.78	26 952	26 431	232.45	2 500	3 000	2 800
KORB200	29 368	ACS+模拟退火	29 369.4	0	29 671	29 421	135.52	1 000	1 500	1 600
		ACS+刻度衰减	29 382.3	0	29 824	29 493	213.54	1 800	2 000	1 900
		ACS	29 837	1.597	30 454	29 951	218.42	2 500	3 000	2 800

(l为循环迭代次数,NC为最大迭代次数). 表1为在ACS中加入动态搜索算子的实验统计结果(EIL50、KORB150、KORA200测试实例,算法运行10次). 可以看出,基于动态搜索策略算法的求解质量和收敛速度均得到了改善,且模拟退火函数$p = p/\log_2(1+l)$比刻度衰减模型$p = p(1 - l/\mathrm{NC})$更具优越性. 加入动态搜索算子后(以EIL50加入模拟退火函数为例),算法只需256次左右的迭代即可收敛,而基本算法需420次迭代才能收敛,故改进算法的全局搜索能力与算法收敛速度的矛盾得到了较好的平衡.

为了验证动态搜索诱导算子通过合理选择p值,可以有效地控制收敛速度,本文采用一组不同的p值进行对比实验,所得结果分别如图1和图2所示.

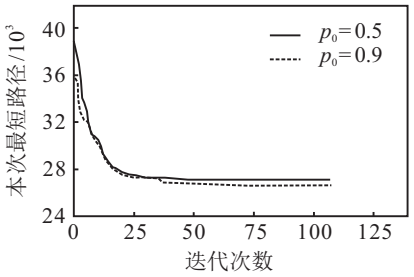


图1　KORB150实例在不同参数ρ值下的平均实验结果

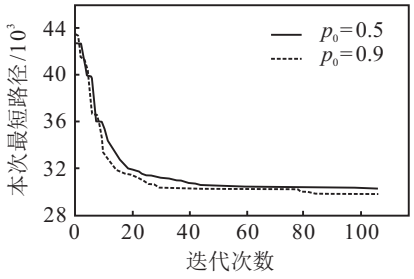


图2　KORA200实例在不同参数ρ值下的平均实验结果

KORB150实验结果表明:采用初始值p较大的算法,其收敛速度比p值较小的算法要快(见图1),即在进化初期采用$p = 0.9$,可以扩大搜索空间,提高种群多样性;而随着进化后期p值的逐渐减少,可以控制搜索偏向信息素增强的方向. 为了更好地说明动态搜索诱导算子的特性,本文还采用更为复杂的KORA200问题进行了测试,实验结果(图2)也表明初期$p = 0.9$时,算法的效果更好.

3.1.2　算法性能比较

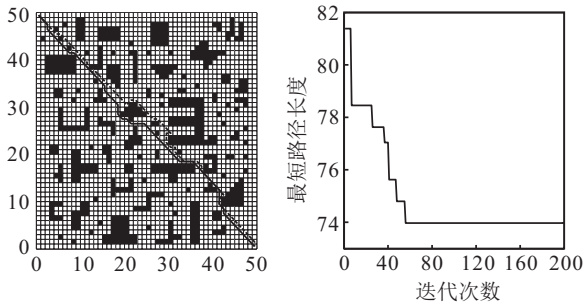
在蚂蚁个数和参数设置相同的情况下,表2对比了本文算法DSACS、ACS+3-opt以及原算法ACS的求解效果,其中算法DSACS采用模拟退火动态搜索算子. 由表2可以说明:在解决较少节点TSP问题(如问题EIL51)时,各算法的搜索能力均很强;而在解决多节点TSP问题(如问题CH144和KORB150)时,本文算法优化解的质量优于对比算法,并且寻找到比标准测试集中最优解更加优化的结果. 表2中标准差的比较表明本文算法具有较好的稳定性(表2中算法误差% = (算法最优解 − 测试集最优解)/测试集最优解 ×100).

3.2　算法及应用研究

将本文算法应用于移动机器人路径规划问题求解. 为了说明算法的优越性,选用文献[12]中改进蚁群算法进行测试. 不失一般性,本文随机测试了一组不同的复杂地图,迭代次数为100次,实线为本文算法寻找到的最优路径,虚线为文献[12]寻找到的最优路径. 从测试得到的结果图3和图4可以看出,本文算法能快速、准确地寻找到最优路径.

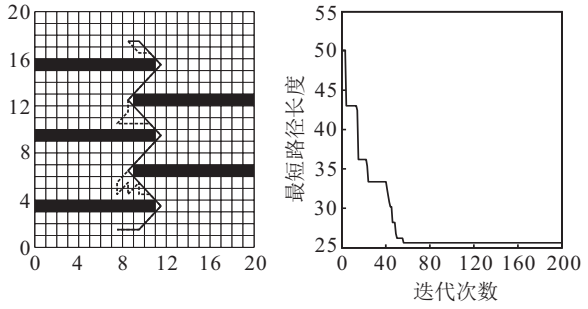
表2　本文算法DSACS、ACS+3-opt以及原算法ACS求解效果比较

TSP问题	测试最优集	比较算法	算法最优解	误差/%	标准差	迭代次数	均值
EIL51	426	DSACS	428.81	0.65	21.671	160	429.38
		ACS+3opt	428.81	0.65	24.178	200	430.21
		ACS	428.98	0.7	27.135	250	431.15
BERLIN52	7 542	DSACS	7 544	0.026	22.75	20	7 548.32
		ACS+3opt	7 544	0.026	62.36	20	7 548.21
		ACS	7 544	0.026	78.25	30	7 550.74
KORA100	21 282	DSACS	21 283.48	0.016	131.22	1 000	21 314.34
		ACS+3opt	21 316.37	0.16	189.31	1 400	21 373.68
		ACS	22 384.64	5.18	236.65	2 000	22 412.76
KORB150	26 130	DSACS	26 127.93	—	122.54	1 500	26 310.27
		ACS+3opt	26 130	0	202.34	2 000	26 341.18
		ACS	26 335	0.78	232.455	2 500	26 570.79
KORA200	29 368	DSACS	29 369	0	135.52	1 000	29 421.43
		ACS+3opt	29 453	0.289	213.54	1 800	29 542.52
		ACS	29 837	1.597	218.42	2 500	29 951.25
PR264	49 135	DSACS	49 138.80	0.009 5	134.57	2 000	49 546.13
		ACS+3opt	49 527.37	0.797	189.72	2 500	49 765.81
		ACS	49 729.52	1.21	234.36	2 500	50 762.45
LIN318	42 029	DSACS	42 178.68	0.36	89.32	2 500	42 212.46
		ACS+3opt	42 435.37	0.96	144.46	2 500	42 521.23
		ACS	42 600.72	1.36	192.65	2 500	42 745.93



(a) 两种算法路径比较 (b) DSACS算法的收敛曲线

图 3 Map50×50时的测试结果



(a) 两种算法路径比较 (b) DSACS算法的收敛曲线

图 4 Map20×20时的测试结果

表3为两种算法实验结果对比(算法运行10次). 与文献[12]改进蚁群算法相比,本文算法在规模较大、环境复杂的情况下,解的质量也显示了较高的稳定性(标准差比较).

表 3 两种算法实验结果对比

算法	Map	优化路径长度	标准差	收敛迭代次数
DSACS	20×20	25.627 4	3.2	56
文献[12]		33.627 4	4.5	72
DSACS	30×30	47.6	4.8	60
文献[12]		51.2	5.7	78
DSACS	50×50	73.982 8	4.3	56
文献[12]		78.225 4	6.8	107

4 结 论

本文提出一种改进的蚁群算法(DSACS)用于解决旅行商问题和移动机器人路径规划问题. 其中:动态搜索诱导算子可以调节种群的多样性并控制算法的搜索方向,从而提高收敛速度;3-opt算子的局域优化能力可以提高优化解的质量. 利用测试实例分析了动态搜索诱导算子的性能,并通过求解机器人路径规划问题验证了算法的实用性. 下一步工作是继续探讨动态搜索诱导算子及数学模型,进一步研究新型蚁群算法模型及性能.

参考文献(References)

[1] Dorigo M, Gambardella L M. Ant colony system: A cooperative learning approach to the traveling salesman problem[J]. IEEE Trans on Evolutionary Computation, 1997, 1(1): 53-66.

[2] 赵娟平,高宪文,刘金刚,等. 移动机器人路径规划的参数模糊自适应窗口蚁群优化算法[J]. 控制与决策, 2011, 26(7): 1096-1100.
(Zhao J P, Gao X W, Liu J G, et al. Parameters self-adaptive fuzzy ant colony optimization algorithm with searching window for path planning of mobile robot[J]. Control and Decision, 2011, 26(7): 1096-1100.)

[3] Silva J L, Nedjah N, Macedo M L, et al. Aco-based static routing for network-on-chips[C]. Computational Science and Its Applications ICCSA-2012. Berlin: Springer, 2012, 7333:113-124.

[4] 金弟,杨博,刘杰,等. 复杂网络簇结构探测——基于随机游走的蚁群算法[J]. 软件学报, 2012, 23(3): 451-464.
(Jin D, Yang B, Liu J, et al. Ant colony optimization based on random walk for community detection in complex networks[J]. J of Software, 2012, 23(3): 451-464.)

[5] Thomas Stutzle T, Holger H Hoos. Max-min ant system[J]. Future Generation Computer Systems, 2000, 16(8): 889-914.

[6] Shuang B, Chen J, Li Z. Study on hybrid PS-ACO algorithm[J]. Applied Intelligence, 2011, 34(1): 64-73.

[7] Deng W, Chen R, He B, et al. A novel two-stage hybrid swarm intelligence optimization algorithm and application[J]. Soft Computing, 2012, 16(10): 1707-1722.

[8] Saenphon T, Phimoltares S, Lursinsap C. Combining new fast opposite gradient search with ant colony optimization for solving travelling salesman problem[J]. Engineering Applications of Artificial Intelligence, 2014, 35: 324-334.

[9] Escario J B, Jimenez J F, Giron-Sierra J M. Ant colony extended: Experiments on the travelling salesman problem[J]. Expert Systems with Applications, 2015, 42(1): 390-410.

[10] 孟祥萍,片兆宇,沈中玉,等. 基于方向信息素协调的蚁群算法[J]. 控制与决策, 2013, 28(5): 782-786.
(Meng X P, Pian Z Y, Shen Z Y, et al. An algorithm based on direction-coordinating[J]. Control and Decision, 2013, 28(5): 782-786.)

[11] Helsgaun K. General k-opt submoves for the Lin-Kernighan TSP heuristic[J]. Mathematical Programming Computation, 2009, 1(2/3): 119-163.

[12] 朱正,刘士荣,张波涛. 一种基于改进蚁群算法的移动机器人全局路径规划方法[C]. 第30届中国控制会议. 烟台, 2011: 4083-4087.
(Zhu Z, Liu S R, Zhang B T. A global path planning method for mobile robot based on improved ant colony algorithm[C]. Proc of the 30th Chinese Control Conf. Yantai, 2011: 4083-4087.)

[13] Stutzle T, Dorigo M. A short convergence proof for a class of ant colony optimization algorithms[J]. IEEE Trans on Evolutionary Computation, 2002, 6(4): 358-365.

[14] Gutjahr J W. ACO algorithms with guaranteed convergence to the optimal solution[J]. Information Processing Letters, 2002, 82(3): 145-153.

(责任编辑:李君玲)