

doi:10.3969/j.issn.1673-9833.2012.04.018

一种快速收敛的自适应蚁群算法

潘伟强, 李长云, 胡盛龙

(湖南工业大学 计算机与通信学院, 湖南 株洲 412007)

摘要: 针对蚁群算法收敛速度慢、参数选择难的不足, 通过分析各参数对算法的影响和比较多种参数寻优方法, 采用粒子群算法对蚁群算法进行参数寻优, 并提出了一种快速收敛的自适应蚁群算法。针对旅行商问题的仿真试验表明, 该算法是可行且有效的。

关键词: 蚁群算法; 自适应; 参数; 旅行商问题

中图分类号: TP301.6

文献标志码: A

文章编号: 1673-9833(2012)04-0076-05

Investigation on a Fast Convergent Adaptive Ant Colony Optimization Algorithm

Pan Weiqiang, Li Changyun, Hu Shenglong

(School of Computer and Communication, Hunan University of Technology, Zhuzhou Hunan 412007, China)

Abstract: The ant colony optimization has deficiencies of slow convergence speed and difficult parameters selection. By analyzing the parameters' effect on the algorithm and comparing multiple parameter optimization methods, adopts the hybrid algorithm of particle swarm optimization and ant colony optimization to optimize parameters, and proposes a fast convergent adaptive ant colony optimization. The simulation of the traveling salesman problem shows that the algorithm is feasible and effective.

Keywords: ant colony optimization; adaptive; parameter; traveling salesman problem

0 引言

1992年, M. Dorigo 在他的博士论文中系统地提出了蚁群算法 (ant colony optimization, ACO), 其灵感来源于蚂蚁在寻找食物过程中发现最优路径的行为。蚁群算法是一种求解复杂优化问题的启发式并行算法。

不管是 M. Dorigo 提出的基本蚁群算法, 还是后来改进的蚁群算法, 都存在与其他启发式算法类似

的不足。本文针对蚁群算法收敛速度慢、参数选择难等不足, 用粒子群算法对蚁群算法进行参数寻优, 并提出了一种快速收敛的自适应蚁群算法。针对旅行商问题的仿真试验表明, 该算法能快速收敛到全局最优解。

1 基本蚁群算法

1.1 基本蚁群算法模型^[1]

n 个城市的对称 TSP 问题 ($0, 1, \dots, n-1$ 表示城市

收稿日期: 2012-06-01

基金项目: 国家技术创新基金资助项目 (11C26214302856), 国家自然科学基金资助项目 (6077311), 中国博士后科学研究基金资助项目 (20080440216)

作者简介: 潘伟强 (1987-), 男, 湖南湘乡人, 湖南工业大学硕士生, 主要研究方向为软件开发技术和智能算法,

E-mail: 821205305@qq.com

通信作者: 李长云 (1971-), 男, 湖南耒阳人, 湖南工业大学教授, 博士, 主要从事可信软件和软件动态演化方面的教学与研究, E-mail: lcy469@163.com

序号)是寻找通过 n 个城市各一次且最后回到出发点的最短路径。该问题存在 $(n-1)!/2$ 条不同路径,随着问题规模 n 的增大,可行解呈指数倍增长,所以采用盲目的穷举算法是不现实的。该问题可以通过模拟蚁群行为进行求解。

首先将 m 只蚂蚁随机放置在 n 个城市,位于城市 i 的第 k 只蚂蚁选择下一个城市 j 的概率为:

$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha(t)\eta_{ij}^\beta(t)}{\sum_{s \in allowed_k} \tau_{is}^\alpha(t)\eta_{is}^\beta(t)}, j \in allowed_k, \\ 0, \text{ otherwise}; \end{cases} \quad (1)$$

$$\eta_{ij}(t) = 1/d_{ij} \quad (2)$$

式中: $\tau_{ij}(t)$ 表示 t 时刻在路径 ij 上残留的信息量; $\eta_{ij}(t)$ 表示启发函数; d_{ij} 表示城市 i 和城市 j 之间的距离; $tabu_k$ 表示记录蚂蚁 k 以前所走过的城市,随着进化过程作动态调整; $allowed_k$ 表示蚂蚁 k 下一步允许选择的城市集。

随着时间的推移,以前留下的信息逐渐消逝,经过 n 个时刻,蚂蚁完成 1 次循环,各路径上信息量要根据式(3)作调整:

$$\begin{cases} \tau_{ij}(t+n) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}, \rho \in (0,1); \\ \Delta\tau_{ij} = \sum_{k=1}^m \Delta\tau_{ij}^k. \end{cases} \quad (3)$$

式中: $\tau_{ij}^k(t)$ 表示第 k 只蚂蚁在本次迭代中留在路径上的信息量; $\Delta\tau_{ij}$ 表示本次循环中路径 ij 信息量的增量。

其中性能较好的蚁周系统模型如式(4)所示:

$$\Delta\tau_{ij} = \begin{cases} \frac{Q}{L_k}, ij \in L_k; \\ 0, \text{ otherwise}. \end{cases} \quad (4)$$

式中: L_k 表示第 k 只蚂蚁在本次迭代中走过的路径; l_k 表示第 k 只蚂蚁在本次迭代的路径长度。

由算法复杂度分析理论可知,算法时间复杂度为 $O(Nc*n^2*m)$,空间复杂度为 $S(n)=O(n^2)+O(n*m)$,其中 Nc 表示循环次数。

1.2 各参数对蚁群算法的影响

蚁群算法的参数是影响其全局求解性能和收敛速度的关键因素,包括信息素挥发因子 ρ 、信息启发式因子 α 、期望启发函数因子 β 、信息素常数 Q 、蚂蚁数目 m 等。因为参数种类的多样性、参数空间的庞大性和参数之间的关联性,所以参数选择较难。詹士昌^[2]、张毅^[3]、陈一昭^[4]等人通过大量的仿真试验,分析了参数 $\alpha, \beta, \rho, m, Q$ 分别对蚁群算法的影响,并确定了最优参数范围,即 $\alpha \in (0, 5), \beta \in (0, 5), \rho \in (0, 1),$

$m \approx n, Q \in (1, 10\ 000)$ 。各参数对算法的影响见表 1。

表 1 参数 $\alpha, \beta, \rho, m, Q$ 对算法的影响

Table 1 The influence of parameters $\alpha, \beta, \rho, m, Q$ on the algorithm

参数	趋势	
	过大	过小
α	选择以前走过路径的概率越大,全局搜索路径的随机性减弱,易陷入局部最优解。	算法正反馈功能降低,且搜索范围减少,陷入局部最优解。
β	算法较容易选择局部最短路径,加快收敛速度,但是随机性不高,容易得到局部最优解。	全局搜索能力加强,但收敛速度慢。
ρ	无效的路径虽然可以被排除搜索,但是不能保证有效的路径被继续搜索,影响最优值的搜索,使得搜索速率降低,并且影响算法的全局搜索能力。	在各路径上残留的信息素过多,导致以前搜索过的路径被选择的概率大,降低全局搜索能力。
m	算法全局搜索能力越强,但信息正反馈作用降低,大量的重复工作消耗了资源,增加了时间复杂度,收敛速度慢。	算法全局搜索能力降低,易得到局部最优解。
Q	算法正反馈功能增强,易得到局部最优解。	算法正反馈功能减弱,全局搜索能力加强,收敛速度慢。

1.3 参数寻优方法

杨亚南^[5]、李世威^[6]等人根据各参数对蚁群算法的影响,提出了采用网格搜索、粒子群算法、遗传算法等方法进行蚁群算法的参数寻优,得到较好的试验结果。

1) 网格搜索。先确定蚂蚁数量 m 、信息素常数 Q 等一般参数,再根据 α, β 等取值范围和规定的水平构造网格,最后用网格交点作为参数组合进行试验。该方法试验可得到参数的优选值。但是该方法具有一定的局限性,这些优选值是所设定的试验所用水平的某种水平,优选值不会超越所取水平的范围。

2) 粒子群(patride swarm optimization, PSO)算法。将蚁群算法的一个参数组合定义为粒子 $p=(\alpha, \beta, \rho)$,用求解问题的目标函数作为适应度函数,把蚁群算法参数优化问题转化为粒子群优化问题。该融合算法步骤简单,能快速地确定蚁群算法的次优参数组合,但是后期粒子群算法局部搜索能力差,很难得到最优参数组合。

3) 遗传(genetic algorithm, GA)算法。将蚁群算法的一个参数组合定义为遗传算法的染色体 $c=(\alpha, \beta, \rho)$,用求解问题的目标函数作为适应度函数,把蚁群算法参数寻优问题转化为遗传算法优化问题。该方法能有效地确定蚁群算法的参数组合,但是解决蚁

群算法参数问题的同时又引入了遗传算法的参数问题,并且该融合算法时间复杂度和空间复杂度较大。

2 自适应蚁群算法

在“探索”和“利用”间寻求一个平衡点是蚁群算法研究的关键问题之一。为了在获得新路径和利用先验知识之间找到平衡点,也就是说既要使得蚁群算法的搜索空间尽可能大,以寻找那些可能存在最优解的解区间;同时,又要充分利用蚂蚁群体内当前所具有的有效信息,使得蚁群算法搜索的侧重点放在那些具有较高适应值的个体所在区间内,从而以较大的概率收敛到全局最优解。在找到全局最优解的前提下,尽可能提高蚁群算法的收缩速度。苏畅^[7]、韦联旺^[8]等人根据不同参数的特点,使用自适应策略对不同模型的蚁群算法进行改进,仿真实验结果表明改进算法是可行的。本文采用自适应的策略来化解求解性能和收敛速度这对主要矛盾。

在蚂蚁寻优过程中,由正反馈作用和先验知识指导蚂蚁的行为:开始时各边被赋予相同的信息素量,蚂蚁主要根据先验知识,相当于贪婪式搜索,即取各边长度的倒数来选择路径;到后期,当各边上的信息素量出现大的差异时,由正反馈作用主导蚂蚁行为,蚂蚁根据信息素量来选择路径。信息素在寻优过程中起到至关重要的作用,故本文先考虑信息素相关的参数,又据文献^[1]得知信息素常量 Q 对算法影响不明显,故不做考虑,即只考虑信息素挥发因子 ρ 和信息素重要程度因子 α 。

2.1 算法改进的原理

针对蚁群算法搜索时间长等不足,提出一种自适应蚁群算法:当搜索停滞时,通过扩大搜索范围,缓解“搜索”与“利用”间的矛盾。算法改进如下:

1) 由于 ρ 的参数取值范围小,对它进行参数微调。当信息素挥发因子 ρ 过小时,在各路径上残留的信息素过多,导致以前搜索过的路径被选择的概率大,全局搜索能力小;当 ρ 过大时,各路径的信息素堆积速度慢,以降低收敛速度为代价增强算法的全局搜索能力。本算法将自适应地修改 ρ ,当算法在 N 次循环内没有改进当前最优解时,进行如下变换:

$$\rho(t+n) = \begin{cases} \varepsilon\rho(t), & \rho(t) \leq \rho_{\max}; \\ \rho_{\max}, & \text{otherwise.} \end{cases} \quad (5)$$

式中: ε 是一个大于1的常数,本文取 $\varepsilon=1.01$; $\rho_{\max}=0.9$ 为信息素挥发因子的最大值,防止其过大时收敛速度降低。

2) 在状态转移公式(1)中,用 $P_{is}^k(\alpha)$ 表示蚂蚁

k 在路径 (i,s) 上信息素重要程度因子为 α 时的状态转移概率,用 $P_{is}^k(\alpha+1)$ 表示蚂蚁 k 在路径 (i,s) 上信息素重要程度因子为 $\alpha+1$ 时的状态转移概率,设 $\tau \in \{\tau_{ij}\}$,

$$\tau^\alpha \times \eta_{is}^\beta / \sum (\tau_{ij}^\alpha \times \eta_{ij}^\beta) = \tau^{\alpha+1} \times \eta_{is}^\beta / \sum (\tau_{ij}^{\alpha+1} \times \eta_{ij}^\beta),$$

化简得: $\tau = \sum (\tau_{ij}^{\alpha+1} \times \eta_{ij}^\beta) / \sum (\tau_{ij}^\alpha \times \eta_{ij}^\beta)$ 。

对于某些城市 s ,若从城市 i 到城市 s ,有 $\tau \leq \tau_{is}$ 则有:

$$\tau_{is} \leq \sum (\tau_{ij}^{\alpha+1} \times \eta_{ij}^\beta) / \sum (\tau_{ij}^\alpha \times \eta_{ij}^\beta);$$

$\tau_{is} (\tau_{is}^\alpha \times \eta_{is}^\beta) / \sum (\tau_{ij}^{\alpha+1} \times \eta_{ij}^\beta) \leq (\tau_{is}^\alpha \times \eta_{is}^\beta) / \sum (\tau_{ij}^\alpha \times \eta_{ij}^\beta)$,
即有

$$P_{is}^k(\alpha+1) \leq P_{is}^k(\alpha)^{[4]}. \quad (6)$$

由以上推导可知:当信息素达到某一临界值后,随着信息素重要程度因子 α 从小变大,则该路径选择的概率由大变小,慢慢地跳出局部最优解,算法的全局搜索能力由弱变强,直到求得全局最优解。上述推导过程可从表1的试验结果中验证其正确性。当算法在 N 次循环内当前最优解没有改进时,根据 α 的参数取值范围和公式(6)的推导过程,进行如下变换:

$$\alpha(t+n) = \begin{cases} \varepsilon_1\alpha(t), & \alpha(t) \leq \alpha_{\max}; \\ \alpha_{\max}, & \text{otherwise.} \end{cases} \quad (7)$$

式中: ε_1 是大于1的常数,本文取 $\varepsilon_1=1.05$; $\alpha_{\max}=5$ 是信息素重要程度因子的最大值,防止蚂蚁以高概率选择以前走过的路径,根据不同的情况, α_{\max} 可以取不同的值。

2.2 算法描述

自适应调整信息素相关参数算法的伪代码如下:

1) 初始化。随机放置蚂蚁,为每只蚂蚁建立禁忌表 $tabu_k$,将初始节点置入禁忌表中。

2) 迭代过程

$k=1$;

while Iter \leq Iter_max do(执行迭代)

for $i=1$ to m do(对 m 只蚂蚁循环)

for $j=1$ to $n-1$ do(对 n 个城市循环)

采用轮盘赌方法和公式1在窗口外

选择下一个城市 j ;

将 j 置入禁忌表,蚂蚁转移到 j ;

end for

end for

计算每只蚂蚁的路径长度;

如果在 N 次循环内当前最优解没有改进,根据公式(5)和(7)更新参数;

根据公式3更新所有蚂蚁路径上的信息素量;

$k=k+1$;

end while

3) 输出结果, 结束算法。

3 TSP 仿真实验

为了检验自适应蚁群算法的求解性能和收敛速度, 进行了仿真试验。试验的硬件平台是CPU为Intel 2.93GHz的PC机, 内存为2.00 GB; 软件平台是Matlab R2009a, 选用文献[1]中的Oliver30问题作为本文的仿真算例, 分别用文献[1]中的基本蚁群算法和本文的自适应蚁群算法进行仿真实验, 并在多项指标上进行比较。

实验采用网格(Grid)搜索、PSO算法、GA算法3种方法对基本蚁群算法和自适应蚁群算法进行参数配置, 参考文献[5-6]编写粒子群和遗传算法参数寻优方法的Matlab程序, 对参数 α, β, ρ 寻优, 其中 $\alpha \in (0, 5.5], \beta \in (0, 5.5], \rho \in [0, 1]$ 。为了使3种参数寻优方法具有可比性, 在网格搜索参数实验中 α, β 步长设为0.5, $\rho \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, 在粒子群和蚁群算法中种群数设为10个、进化代数为12代。这样每组实验调用蚁群算法求个体适应度的次数都为120次。3种方法每组实验各运行10次, 实验结果见表2和图1~2。

表2 三种参数配置方法的比较

Table 2 Comparison of three parameters configuration methods

优化算法	最优解	最优解总次数	平均运行时间 /s
Grid	423.740 6	23	2 082.5
PSA	423.740 6	58	2 080.2
GA	423.740 6	52	2 491.6

由表2可知, 3种寻优算法都能找到全局最优解及对应的参数组合, 10次试验得到全局最优解的次数从多到少依次是PSO, GA, Grid, 程序运行所花费的平均时间从少到多依次是PSO, Grid, GA。

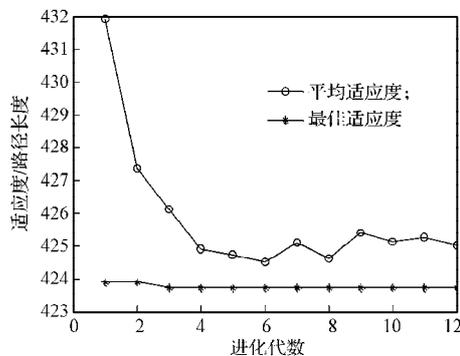


图1 粒子群算法参数寻优的进化曲线

Fig. 1 The evolution curve of parameter optimization of particle swarm algorithm

图1和图2分别为粒子群算法的进化曲线与遗传算法的进化曲线, 从图1中可以看到代表种群适应度的平

均适应度(平均路径长度)曲线朝最佳适应度曲线靠近; 而图2中平均适应度曲线有接近最佳适应度的时候, 在11代之后出现返祖现象, 说明这时候种群没有朝最优方向进化; 网格搜索寻优只能在事先定义好的区间和水平内搜索, 其搜索空间小, 不利于参数寻优。综上所述, 粒子群算法在蚁群算法参数优化方面比其他2种方法好。

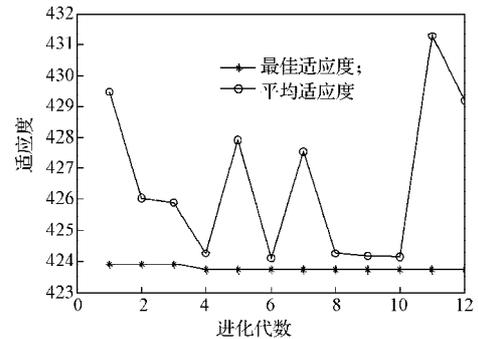


图2 遗传算法参数寻优的进化曲线

Fig. 2 The evolution curve of parameter optimization of genetic algorithm

根据上述讨论, 采用粒子群算法对自适应蚁群算法进行参数寻优, 编写相应的Matlab程序, 运行之后得到自适应蚁群算法最优参数组合及相应结果见表3, 其中参数组合表示参数的初值。自适应蚁群算法和基本蚁群算法在上述算例中的其他参数设置为 $m=30, Q=10$, 最大迭代次数 $N_c=1\ 000$ 。由于蚁群算法本身的随机性, 每个参数寻优试验重复运行10次, 选取合适的样本, 统计出科学的试验结果。

表3 自适应蚁群算法和基本蚁群算法的性能比较

Table 3 The performance comparison of adaptive ant colony optimization and basic ant colony optimization

蚁群算法	参数组合(α, β, ρ)	最优解迭代次数	平均迭代次数
ACO	(0.342 41, 5.500 00, 0.320 68)	498	571.00
	(0.468 70, 4.852 30, 0.256 77)	864	
	(0.443 04, 2.678 70, 0.862 17)	258	
	(0.321 51, 5.500 00, 0.807 46)	562	
	(0.422 45, 4.329 30, 0.153 98)	673	
SACO	(0.851 08, 3.670 90, 0.358 66)	16	50.20
	(0.851 07, 1.829 10, 0.311 79)	41	
	(0.523 45, 3.670 90, 0.358 66)	69	
	(0.851 07, 1.829 10, 0.311 79)	103	
	(0.851 07, 3.670 90, 0.358 66)	22	

由表3可知, 基本蚁群算法和自适应蚁群算法收敛到全局最优解的最少迭代次数分别是258和16次, 而文献[1]中的最好迭代次数是342次; 多个样本的平均迭代次数分别是571.00和50.20次。因此, 自适应蚁群算法比基本蚁群算法更快收敛到全局最优解。

图3表示自适应蚁群算法的进化曲线,波动较大的线表示蚁群在各迭代中的平均距离,随着迭代次数的增加,蚁群有朝有序方向变化的迹象;波动较小的线表示蚁群在各迭代中的最短距离,随着迭代次数的增加,曲线呈阶梯形变化,这是由于在 N 次迭代中如果当前最优解没有明显变化时,算法会根据各参数对算法的作用产生自适应参数,扩大寻优范围,跳出局部最优解,直至找到全局最优解为止。图4是基本蚁群算法的进化曲线,其全局最优解长时间没有变化。由此可以看出自适应蚁群算法进化所需的迭代次数明显比基本蚁群算法少,前者的收敛速度比后者快;固定参数与自适应参数相比,前者不利于蚁群算法快速地收敛到全局最优解。仿真实验表明,本文所提出的算法能快速收敛到全局最优解。

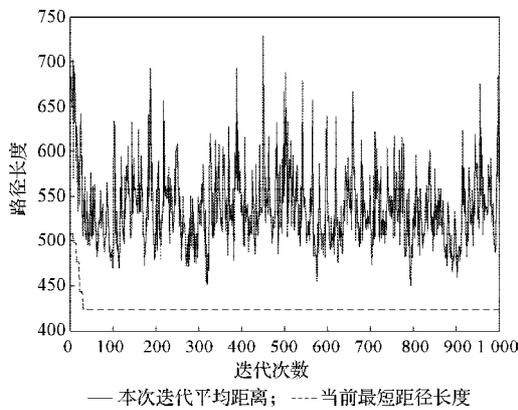


图3 自适应蚁群算法的进化曲线

Fig. 3 The evolution curve of the adaptive ant colony optimization

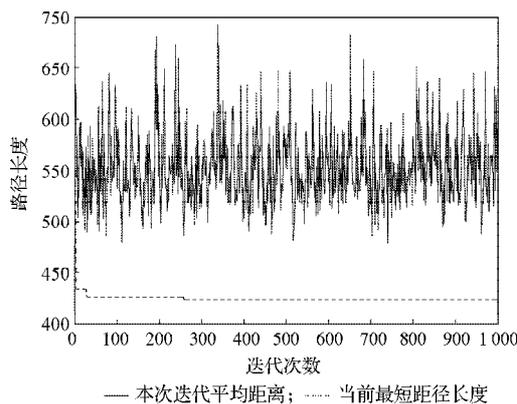


图4 基本蚁群算法的进化曲线

Fig. 4 The evolution curve of the basic ant colony optimization

4 结语

蚁群算法是一种求解复杂优化问题的启发式并行算法,存在参数选择难、收敛速度慢等不足。本文用粒子群算法对其进行参数优化,并提出了一种快速收敛的自适应蚁群算法:如果该算法当前最优

值没有明显变化时,算法就自适应调整信息素相关参数,扩大搜索范围,使算法跳出局部最优解,直至找到全局最优解。大量仿真试验表明,本算法是可行且有效的。

参考文献:

- [1] Dorigo M, Maniezzo V, Coloni A. The Ant System: Optimization by a Colony of Cooperating Agents[J]. IEEE Transactions on Systems Man and Cybernetics-Part B, 1996, 26(1): 29-41.
- [2] 詹士昌,徐婕,吴俊. 蚁群算法中有关算法参数的最优选择[J]. 科技通报, 2003, 19(5): 381-386.
Zhang Shichang, Xu Jie, Wu Jun. The Optimal Selection on the Parameters of the Ant Colony Algorithm[J]. Bulletin of Science And Technology, 2003, 19(5): 381-386.
- [3] 陈一昭,姜麟. 蚁群算法参数分析[J]. 科学技术与工程, 2011, 11(36): 9081-9084.
Cheng Yizhao, Jiang Lin. Parametric Study of Ant Colony Optimization[J]. Science Technology and Engineering, 2011, 11(36): 9081-9084.
- [4] 张毅,梁艳春. 蚁群算法中求解参数最优选择分析[J]. 计算机应用研究, 2007, 24(8): 70-71.
Zhang Yi, Liang Yanchun. Research of Optimal Selection on Parameters of Ant Colony Algorithm[J]. Application Research of Computers, 2007, 24(8): 70-71.
- [5] Yang Yanan, You Jing. Optimizing Parameter of Ant Colony Algorithm Based on Particle Swarm Algorithm[C]//3rd International Conference on Computational Intelligence and Industrial Application. Wuhan: [s.n.], 2010: 246-249.
- [6] 李世威,王建强,曾俊伟. 遗传算法调整蚁群算法参数模型研究[J]. 计算机工程与设计, 2011, 32(10): 3490-3493.
Li Shiwei, Wang Jianqiang, Zeng Junwei. Model of Ant Colony Algorithm Parameters Optimization Based on Genetic Algorithm[J]. Computer Engineering and Design, 2011, 32(10): 3490-3493.
- [7] 苏畅,徒君. 一种自适应最大最小蚁群算法[J]. 模式识别与人工智能, 2007, 20(5): 688-691.
Su Chang, Tu Jun. An Adaptive Max-Min Ant Colony Algorithm[J]. Pattern Recognition and Artificial Intelligence, 2007, 20(5): 688-691.
- [8] 韦联旺,段复建. 带参数信息素的蚁群算法[J]. 桂林电子科技大学学报, 2011, 31(2): 155-159.
Wei Lianwang, Duan Fujian. Ant Colony Algorithm with Parameter Pheromone[J]. Journal of Guilin University of Electronic Technology, 2011, 31(2): 155-159.

(责任编辑:申剑)