第44卷 第2期 西 安 交 通 大 学 学 报 Vol. 44 №2
2010年2月 JOURNAL OF XI′AN JIAOTONG UNIVERSITY Feb. 2010

采用序优化的改进蚁群算法

张兆军[1,2]，冯祖仁[1,2]，任志刚[1,2]
(1. 西安交通大学系统工程研究所，710049，西安；2. 西安交通大学机械制造系统工程国家重点实验室，710049，西安)

摘要：为了评价蚁群算法在有限时间内所得优解的质量，基于序优化方法提出了一种改进的蚁群算法：使用盲目挑选规则选择初始解，并对信息素进行相应的初始化；确定得到满足要求的优解所需要的迭代次数，将其作为算法的终止条件；为了更好地利用每次迭代中的优解，在算法开始阶段使用前 l 个迭代优解更新信息素，以增强探索能力；在算法结束阶段采用当前迭代最优解更新信息素，以加快收敛速度. 改进算法在保证收敛的前提下，并没有增加算法的时间复杂度. 对旅行商问题进行的仿真实验表明，改进算法在解的质量和收敛速度方面优于最大-最小蚂蚁系统.

关键词：蚁群算法；序优化；盲目挑选；旅行商问题
中图分类号：TP18 **文献标志码**：A **文章编号**：0253-987X(2010)02-0015-05

Novel Ant Colony Optimization Algorithm Based on Order Optimization

ZHANG Zhaojun[1,2], FENG Zuren[1,2], REN Zhigang[1,2]
(1. Systems Engineering Institute, Xi′an Jiaotong University, Xi′an 710049, China; 2. State Key Laboratory for Manufacturing Systems Engineering, Xi′an Jiaotong University, Xi′an 710049, China)

Abstract: To evaluate the quality of optimal solutions obtained by the ant colony optimization (ACO) algorithm in limited time, an improved ACO algorithm is presented on the basis of the ordinal optimization. An initial solution is selected using the blind picking rule, and the pheromone is initialized correspondingly. The number of iterations to achieve the optimal solution meeting the demand is then determined and is used as the termination condition of the algorithm. To make better use of the solutions obtained at each iteration, the first l solutions are employed to enhance search capability at the beginning phase of the algorithm. While the current optimal solution is used at the end phase of the algorithm to accelerate the convergence. The time complexity of the novel algorithm is not increased under the condition that ensures the convergence. Simulation results on the traveling salesman problem show that the proposed algorithm is superior to the max-min ant system in both the quality of solutions and the speed of convergence.

Keywords: ant colony optimization; ordinal optimization; blind picking; traveling salesman problem

蚁群算法[1]是一种仿生随机优化算法，已被成功应用于旅行商问题(TSP)、二次分配、网络路由、属性约简[2]等问题的求解，具有鲁棒性、正反馈、分布式计算和易与其他算法结合等优点. 然而，现有方法也存在一些不足，如初期搜索时间偏长，容易陷入局部最优解等. 为此，学者们提出了很多改进算法，例如使用局部更新策略和全局更新策略的蚁群系统[3]，限制信息素的上、下界并使用最优解更新策略的最大-最小蚂蚁系统(max-min ant system, MMAS)[4]等. 此外，文献[5]受神经网络和遗传算法的启发，提出了一种二进制蚁群进化算法；文献[6]将分散搜索的思想融入蚁群算法，提高了算法的

收稿日期：2009-06-20. 作者简介：张兆军(1981—)，男，博士生；冯祖仁(联系人)，男，教授，博士生导师. 基金项目：国家自然科学基金资助项目(60875043)；国家重点基础研究发展规划资助项目(2007CB311006).

全局搜索能力.

虽然蚁群算法在模型改进、拓展应用等方面取得了一系列成果,但是仍然存在一些问题有待进一步研究,其中一个问题是评价蚁群算法所得解的质量. 收敛性分析表明,当搜索时间无限长时,某些蚁群算法能够以概率 1 找到全局最优解. 然而,在实际计算中时间是有限的,因此评价有限时间内蚁群算法所得优解的质量具有实际意义.

作为一种随机优化算法,由哈佛大学何毓琦教授提出的序优化方法[7]能够从概率上对优化结果进行定量评估,理论体系比较完善[8],适合作为许多优化算法的补充[9-10]. 本文基于序优化理论,提出一种改进的蚁群算法,以解决蚁群算法所得优解质量的评价问题.

1 最大-最小蚂蚁系统和序优化方法简介

1.1 最大-最小蚂蚁系统

由 Stützle 等[4]提出的最大-最小蚂蚁系统(MMAS),是目前使用比较多的一种改进蚁群算法. 在蚁群系统(ant system,AS)的基础上,MMAS 主要有以下几方面的改进:①只允许 1 只最优蚂蚁进行信息素更新;②为了避免算法的早熟现象,把信息素设置在一个区间$[\tau_{\min},\tau_{\max}]$内;③信息素的初始值设定为信息素的上界 $\tau_{\max}$.

为了更好地理解 MMAS,下面以 TSP 为例来描述 MMAS 的基本步骤[4]:首先,初始化参数;然后,每只蚂蚁根据路径上的信息素浓度,按照路径选择策略,构造 TSP 的解;当所有蚂蚁完成一次循环后,按照信息素更新策略更新信息素;最后,当达到算法终止条件时,算法终止.

路径选择策略为:假设 t 次迭代蚂蚁 k 在第 i 个城市,按照式(1)计算选择路径(i,j)的概率

$$P(j\mid i)=\begin{cases}\dfrac{\tau(i,j)^{\alpha}\eta(i,j)^{\beta}}{\sum\limits_{r\in N_i^k}\tau(i,r)^{\alpha}\eta(i,r)^{\beta}} & \text{如果 } j\in N_i^k\\ 0 & \text{其他}\end{cases}\tag{1}$$

式中:α、β 是参数;$\tau(i,j)$表示路径(i,j)上的信息素;N_i^k是与 i 相连的所有城市;r 是 N_i^k中的某个城市;$P(\cdot\mid\cdot)$表示条件概率;$\eta(i,j)$表示路径(i,j)上的启发信息.

常用的信息素更新策略为:当所有蚂蚁都获得一个完整路径后,算法按照下式更新信息素

$$\tau_{ij}(t+1)=\left[(1-\rho)\tau_{ij}(t)+\Delta\tau_{ij}^{\text{best}}(t)\right]_{\tau_{\min}}^{\tau_{\max}}\tag{2}$$

$$[x]_b^a=\begin{cases}a & \text{若 } x>a\\ b & \text{若 } x<b\\ x & \text{其他}\end{cases}\tag{3}$$

式中:$\rho(0<\rho\leqslant 1)$表示信息素挥发系数;$\tau_{\max}$和 $\tau_{\min}$分别表示信息素的上界和下界;$\Delta\tau_{ij}^{\text{best}}$表示路径$(i,j)$上的信息素增量. 令 S^{best}表示用信息素更新的最优解,$\Delta\tau_{ij}^{\text{best}}$定义为

$$\Delta\tau_{ij}(t)=\begin{cases}\dfrac{1}{f(S^{\text{best}})} & \text{若边出现在 } S^{\text{best}} \text{ 中}\\ 0 & \text{其他}\end{cases}\tag{4}$$

式中:$f(S^{\text{best}})$表示最优解 S^{best}对应的路径长度,S^{best}可能的取值是 $S_{\text{ib}}^{\text{best}}$或 $S_{\text{bs}}^{\text{best}}$,$S_{\text{ib}}^{\text{best}}$表示当前迭代最优解,$S_{\text{bs}}^{\text{best}}$表示至今迭代最优解.

1.2 序优化方法

序优化方法有 2 个基本思想:第一,序比较,因为“序”比“值”更易确定,故用序比较代替值比较;第二,目标软化,即通过目标软化放松优化目标[8]. 序优化从工程实际出发,以寻找足够好的解为目标,进而减少搜索量.

假设在搜索空间内定义一个足够好子集 G,大小是 g,可以是样本集的前 g 个解,或是解的 top-p. 基于一定挑选规则选取另外一个子集 S,称为选择子集,大小是 s. 由序优化理论可以计算出 $G\cap S$ 至少有 k 个的置信概率 P_{AP}.

恰当地选择挑选规则能够有效地节省计算量,提高置信概率. 盲目挑选(blind picking,BP)规则是最基本的挑选规则,常被用来衡量其他挑选规则的效率. 本文提出的改进算法中使用 BP 规则进行初始解的选取和终止条件的确定.

2 基于序的蚁群算法

2.1 问题分析

蚁群算法是一种元启发式算法,虽然在无限时间下该方法是收敛的,但是由于缺乏系统的分析方法,目前关于算法收敛时间的研究尚处于起步阶段,对有限时间所得优解的评价就更加困难. 目前,优解评价的常用方法是传统的比值比较,需要知道已知最优解,但这在实际中是很难得到的,另外蚁群算法是按照概率选择构造解,参数之间的耦合性很强,分析各个参数对解的影响有很大的困难,这些都是优解评价中的难题.

进一步分析蚁群算法发现,在每次迭代中都有对迭代解的排序比较,这和序优化的思想类似,而序

优化方法的解评价理论比较完善，从而为启发式算法的优解质量评价研究提供了借鉴. 另外，在信息素更新中只使用最优解的更新虽然有利于充分利用每次迭代产生的最优解，但是不利于解空间的探索，如果所有解参与信息素更新(在 AS 中就是采用这种方式)，则又影响收敛速度，因此需要考虑一种更为有效的信息素更新方式.

2.2　改进策略

基于序优化的改进蚁群算法主要在 3 个方面做了改进，具体描述如下.

(1)以一定概率选取初值，并进行信息素初值的设置. 以序优化理论中的 BP 规则为基础，按照一定的置信概率选择初始解，并设置信息素初值.

BP 规则[8]为：令 P_{AP} 表示置信概率，$p=g/N$ 表示足够好解占搜索空间的比例，$s=|S|$ 表示随机抽样次数，则有

$$P_{AP}(|S\cap G|\geqslant 1)\geqslant 1-(1-p)^s \tag{5}$$

又因为 $1-x\leqslant e^{-x}$，有 $P_{AP}(|S\cap G|\geqslant 1)\geqslant 1-e^{ps}$，两边取对数，可以得到满足置信概率 P_{AP} 的足够好解 p 所需要的抽样次数

$$s\leqslant\frac{-\ln(1-P_{AP})}{p} \tag{6}$$

一般 P_{AP} 和 p 是预先指定的，这样就可以得到所需要的抽样次数.

(2)终止条件的确定. 根据 BP 规则计算出以一定置信概率得到满足要求的足够好解所需要的迭代次数，作为蚁群算法的终止条件. 这与一般使用经验确定的终止条件(如算法运行到最大迭代次数或算法陷入了停滞状态)相比，不仅更有理论依据，而且能预先估计所得优解的质量.

(3)信息素的更新方式. 在实验中发现，有些解的质量虽然比最优解的质量稍差，可是仍含有一些有用信息，合理利用这些信息将有助于找到最优解. 尤其是在算法初期，解空间的探索越充分，后期得到的优解的质量就越高. 但是，若所有解均参与信息素更新，又将导致收敛速度过慢. 因此，在信息素更新策略中只使用部分迭代优解更新信息素，称之为按序更新策略.

按序更新策略：将 t 次迭代的解排序，使用前 l 个迭代优解根据式(7)更新信息素

$$\tau_{ij}(t+1)=\left[(1-\rho)\tau_{ij}(t)+\sum_{\omega=1}^{l}\Delta\tau_{ij}^{\omega}(t)\right]_{\tau_{\min}}^{\tau_{\max}} \tag{7}$$

式中：$\sum_{\omega=1}^{l}\Delta\tau_{ij}^{\omega}(t)$ 表示 l 只蚂蚁在路径(i,j)之间根据排名对信息素进行更新；

$$\Delta\tau_{ij}^{\omega}(t)=\begin{cases}\dfrac{1}{\omega f(S^{\omega})} & \text{若第 }\omega\text{ 只最好的蚂蚁经过该边}\\ 0 & \text{其他}\end{cases} \tag{8}$$

表示由第 ω 只最好的蚂蚁引起的路径(i,j)上的信息素增量，其中 ω 表示解 S^{ω} 在该次迭代解中排列的顺序号，$f(S^{\omega})$表示第 ω 只最好的蚂蚁对应的路径长度. 顺序越靠前，所获得的信息素更新量就越大.

实验中发现，如果只使用按序更新策略来更新信息素(与文献[11]提出的 AS_{rank} 算法有些类似)，则所得解的质量不如 MMAS. 因此，本文综合考虑各个因素，使用更为灵活的方式来更新信息素. 令 N_{max} 表示最大迭代次数，m_1 是介于 0 和 1 之间的实数，N_C 表示迭代次数. 在算法的开始阶段，即 $N_C\leqslant m_1N_{max}$ 时，每次迭代中将所得的解按照构造路径的长度从小到大进行排序，采用排在前的 l 个当前迭代优解进行信息素更新，称之为按序更新策略；在算法结束阶段，即 $N_C>m_1N_{max}$ 时，使用当前迭代最优解进行信息素更新，目的是加快算法的收敛速度.

2.3　仿真步骤和分析

下面以 TSP 为例，描述本文算法的主要步骤.

第 1 步：初始化参数，其中包括 α、β、ρ、l、m_1 等.

第 2 步：确定终止规则. 根据式(6)计算以置信概率 P_{AP1} 得到足够好解 p_1 所需要的最大迭代次数，作为算法的终止条件.

第 3 步：选取初始解. 根据式(6)，确定满足置信概率 P_{AP2} 的足够好解 p_2 作为初值，并对信息素矩阵进行相应的设置，初始化为最大值.

第 4 步：按照路径选择策略，构造 TSP 的解.

第 5 步：分阶段更新信息素. 如果 $N_C\leqslant m_1N_{max}$，则按照式(7)，使用每次迭代的前 l 个优解进行按序更新；否则，按照式(2)进行更新.

第 6 步：判断终止条件是否满足. 若满足，算法终止；否则，返回到第 4 步.

在第 5 步中，信息素更新过程包括判断信息素是否超出了上、下界. 上、下界按照文献[4]中的公式确定.

下面分析本文算法的时间复杂度. 令 n 为 TSP 中城市的数量，m 为算法中使用的蚂蚁数，则第 1 步的时间复杂度为 $O(n^2+m)$；第 4 步的时间复杂度为 $O(n^2m)$；第 5 步中虽然信息素更新是分阶段进行的，但是总体时间复杂度应该小于 AS 算法的 $O(n^2)$；第 6 步的时间复杂度为 $O(nm)$，由于在第 3

步中使用了 BP 规则，故其时间复杂度是 $O(n^2 N_C)$. 虽然 BP 规则对时间复杂度有所影响，但是由于在整个算法中，只在开始使用了 1 次 BP 规则确定初始解，因此并不影响整个算法的整体时间复杂度. 最终，改进蚁群算法的时间复杂度为 $O(N_{max}\, n^2\, m)$，这和基本蚁群算法的时间复杂度是一样的. 因此，本文算法并没有增加原来算法的时间复杂度.

本文算法中有明显的信息素下界的限制，因此本文算法属于 $ACO_{\tau_{min}}$ 算法，而文献[12]表明，$ACO_{\tau_{min}}$算法是收敛的，所以本文算法也是收敛的. 由于信息素更新方式的不同，在问题规模较小的情况下，本文算法的收敛速度要稍慢于 MMAS，但是随着问题规模的增大，本文算法的平均收敛速度要快于 MMAS，而且获得的解的质量要高于 MMAS. 特别地，当 $m_1=1$、$l=1$ 时，本文算法与 MMAS 算法类似；当 $m_1=1$、$l>1$ 时，本文算法与 AS_{rank}算法类似.

3 仿真实验

为了验证本文所提改进算法的有效性，将其应用于 TSP，用 Matlab 实现了 MMAS 和本文算法，运行环境为 PIV 2.0 GHz CPU，1 GB RAM，Windows XP 操作系统. 测试算例选自 TSPLIB，算例名称中的数字表示城市数(kro124p 是个例外，它的城市数是 100). 本文算法中，参数 $\alpha=1$，$\beta=2$，初始解的挑选是以 $P_{AP2}=99\%$挑选前 $p_2=0.1\%$的最优解，由式(6)得到需要的抽样次数是 4 605.

3.1 参数设置

本文算法中的另外 2 个参数是 l 和 m_1，下面分析它们对本文算法性能的影响. 实验中只改变一个参数，另一个参数保持不变. 测试中蚂蚁数 $m=20$，最大迭代次数 $N_{max}=2\,000$，每个算例独立测试 10 次，所得结果如表 1 和表 2 所示. 由表 1 和表 2 可以看出，m_1 和 l 的取值不能过大或者过小. 当 l 过大时，算法与 AS 算法类似；当 m_1 过大时，算法则与 AS_{rank}算法类似. 在相同条件下，这 2 种算法的性能都比 MMAS 差，这与文献[4]中的结果是一致的. 参数 m_1 和 l 要综合最优值和平均值等方面的信息后确定.

3.2 实验结果比较

终止条件是 $P_{AP1}=99.99\%$和 $p_1=0.01\%$，由式(6)得到迭代次数为 92 103. 为了方便比较，在算法中令蚂蚁数 $m=31$，最大迭代次数 $N_{max}=3\,000$，其他参数 $\alpha=1$，$\beta=2$，$m_1=0.3$，$l=10$，每个算例独立测试 25 次. MMAS 中的参数设置参见文献[4].

测试结果如表 3 所示(其中前 3 个算例是对称 TSP，后 3 个算例是非对称 TSP). 从表 3 可以看出，对于各种 TSP 算例，无论是最优路径长度还是平均路径长度，本文的改进算法均优于 MMAS，而且标准差更小，尤其是在 eil51、kroA100 和 ry48pt 算例中均取得已知最优解. 除 eil51 算例外，本文算法的平均迭代次数均少于 MMAS 的，说明本文算法的收敛速度快于 MMAS 的收敛速度. 为更直观地描述算法的迭代过程，以 kroA100 为例描述了这 2 种算法的进化曲线，如图 1 所示. 为了使结果更加直观以利于比较，在不影响最终结果的前提下，图 1 中省去了前 300 次迭代的比较. 从图 1 可以看出，本文算法在 900 次迭代以前所获得的优解质量比 MMAS 的要差，这是因为本文算法在这个阶段主要是强调对解空间的探索，而在 900 次迭代以后，随着本文算法进入第二个阶段，更加强调了对解空间的开发，使得最终获得的优解质量要高于 MMAS 的优解质量.

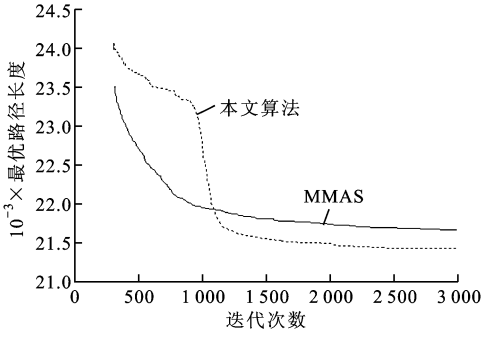

图 1 本文算法与 MMAS 的进化曲线

表 1 $m_1=0.2$、l 取不同值时所得实验结果的对比

TSP 算例	$l=5$		$l=10$		$l=15$		$l=20$	
	最优路径长度	平均路径长度	最优路径长度	平均路径长度	最优路径长度	平均路径长度	最优路径长度	平均路径长度
ry48p	14 507	14 588.0	14 481	14 560.3	14 422	14 617.9	14 612	14 828.6
eil51	426	428.8	427	430.7	428	430.9	427	431.9

表 2　$l=10$、m_1 取不同值时所得实验结果的对比

TSP 算例	$m_1=0.1$		$m_1=0.3$		$m_1=0.5$		$m_1=1.0$	
	最优路径长度	平均路径长度	最优路径长度	平均路径长度	最优路径长度	平均路径长度	最优路径长度	平均路径长度
ry48p	14 507	14 686.9	14 507	14 637.1	14 516	14 649.8	14 714	15 044.1
eil51	428	430.8	426	432.1	427	432.6	449	454.1

表 3　本文算法与 MMAS 的实验结果比较

TSP 算例	算法名称	最优路径长度	最差路径长度	平均路径长度	标准差	平均迭代次数
eil51	本文算法	426(0.00%)	436	428.7	2.322 4	1 445.2
	MMAS	427(0.23%)	438	430.6	3.489 0	1 165.2
kroA100	本文算法	21 282(0.00%)	22 067	21 449.6	190.954 9	1 982.0
	MMAS	21 305(0.11%)	22 523	21 664.1	304.732 5	1 993.1
d198	本文算法	16 002(1.41%)	16 454	16 172.6	180.288 0	2 667.7
	MMAS	16 027(1.57%)	17 020	16 262.7	207.764 2	2 766.2
ry48p	本文算法	14 422(0.00%)	14 914	14 627.4	140.938 6	1 699.8
	MMAS	14 446(0.17%)	15 161	14 691.2	189.327 2	1 827.8
ft70	本文算法	39 285(1.58%)	40 363	39 866.1	379.240 2	2 531.9
	MMAS	39 377(1.82%)	41 113	40 146.0	427.591 3	2 561.4
kro124p	本文算法	36 990(2.10%)	39 005	37 858.4	609.159 4	2 398.3
	MMAS	37 369(3.14%)	40 300	38 586.3	715.803 0	2 617.8

注：括号中的数值为最优路径长度与已知最优解的相对误差.

4　结　论

基于序优化方法，本文提出了一种能够评价优解质量的改进蚁群算法，详细讨论了在初始解的选取、终止条件的确定和信息素更新等方面所做的改进工作. 本文算法在初始化阶段以置信概率 P_{AP} 保证至少得到一个 top-p 的解，使用按序更新策略可以获得更多解空间的信息，进一步提高了算法所得解的质量. 最后，由 BP 规则确定终止条件，可以使本文算法在一定的计算量下，以高置信概率找到足够好的解. 仿真实验结果表明，本文所提改进算法是有效的.

参考文献：

[1] DORIGO M, MANIEZZO V, COLORNI A. The ant system: optimization by a colony of cooperating agents [J]. IEEE Trans on Systems, Man, and Cybernetics: B, 1996, 26(1): 29-41.

[2] 任志刚，冯祖仁，柯良军. 蚁群优化属性约简算法 [J]. 西安交通大学学报，2008，42(4)：440-444.
REN Zhigang, FENG Zuren, KE Liangjun. Ant colony optimization approach to attribute reduction problem [J]. Journal of Xi′an Jiaotong University, 2008, 42(4): 440-444.

[3] 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.

[4] STÜTZLE T, HOOS H H. Max-min ant system [J]. Future Generation Computer Systems, 2000, 16(9): 889-914.

[5] 熊伟清，魏平. 二进制蚁群进化算法 [J]. 自动化学报，2007，33(3)：259-264.
XIONG Weiqing, WEI Ping. Binary ant colony evolutionary algorithm [J]. Acta Automatica Sinica, 2007, 33(3): 259-264.

[6] 张晓霞，唐立新. 一种求解 TSP 问题的 ACO & SS 算法设计[J]. 控制与决策，2008，23(7)：762-766.

[7] HO Y C, SREENIVA R S. Ordinal optimization of

discrete event dynamic systems [J]. Journal of DEDS, 1992, 2(2): 61-88.

[8] HO Y C. An explanation of ordinal optimization: soft computing for hard problems [J]. Information Sciences, 1999, 113(3/4): 169-192.

[9] 张亮，王凌，郑大钟. 有限计算量下模拟退火算法的参数序优化 [J]. 控制与决策，2004，19(2)：226-229.
ZHANG Liang, WANG Ling, ZHENG Dazhong. Parameter ordinal optimization for simulated annealing with limited computational efforts [J]. Control and Decision, 2004, 19(2): 226-229.

[10] MORI H, TANI H. A hybrid method of PTS and ordinal optimization for distribution system service restoration[C]. IEEE International Conference on Systems, Man and Cybernetics. Piscataway, NJ, USA: IEEE, 2003: 3476-3483.

[11] BULLNHEIMER B, HARTL R F, STRAUSS C. A new rank-based version of the ant system: a computational study [J]. Central European Journal for Operations Research and Economics, 1999, 7(1): 25-38.

[12] STÜTZLE 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.

（编辑 葛赵青 苗凌）