

基于交互式的并行蚁群优化算法^①

孟 鑫, 安毅生, 张志明

(长安大学 信息工程学院, 西安 710064)

摘 要: 传统蚁群优化算法研究已经取得了很重要成果, 但是在解决大规模组合优化问题时仍存在早熟收敛, 搜索时间长等缺点. 为此, 将邻域搜索技术与蚁群优化算法进行融合, 提出一种新的并行蚁群优化算法, 实验结果表明, 在解决大规模 TSP 问题时, 该算法求解质量和稳定性更好, 在短时间内即可得到高质量的解.

关键词: 组合优化问题; 邻域搜索技术; 并行蚁群优化算法; 邻域搜索; TSP 问题

Interactive-Based Parallel Ant Colony Optimization

MENG Xin, AN Yi-Sheng, ZHANG Zhi-Ming

(College of Information Engineering, Chang'an University, Xi'an 710064, China)

Abstract: Although a lot of important results are achieved in the research of traditional ant colony optimization, and there are many shortcomings in solving large-scale combination optimization problems, such as premature convergence and time consuming. Therefore, a parallelization of ant colony algorithm based on the combination of neighborhood search algorithm and ant colony optimization is proposed and realized. The results of experiment show that the parallel algorithm has much higher quality and stability than that of traditional serial ant colony optimization for solving large-scale TSP problems.

Key words: combination optimization problems; neighborhood search algorithm; parallel ant colony optimization; neighborhood search; TSP problems

随着工程科学的飞速发展, 实际工程应用中需要解决的问题规模也越来越大, 借助计算机的优越性能进行求解必不可少, 而传统单一处理器的计算速度已经不能满足日益庞大的计算任务, 越来越多的研究者开始关注并行计算. 与此同时, 硬件技术的成熟以及处理器和网络性能不断提升, 已经使并行计算完全有能力脱离昂贵的超级计算平台从而移植到普通计算机上, 毫无疑问, 并行计算将逐渐成为解决大规模工程优化问题的最佳选择.

旅行商问题(Travelling Salesman Problem, TSP)是数学领域中一个典型的优化组合问题, 并且是一个 NP 难问题, 也是许多工程优化问题的抽象形式. TSP 问题主要分为以下几类: 对称旅行商问题; 非对称旅行商问题, 即对于任意两城市 i 和 j 间的距离存在 $d_{ij} \neq d_{ji}$; 多目标旅行商问题, 在普通 TSP 问题中只有一个权重即距

离, 而对于多目标旅行商问题, 城市间的权重有多个(比如距离、费用等), 需要求得在访问完所有城市回到初始城市后的总权重值最小. 针对 TSP 这类从工程领域抽象出来的组合优化问题, 如果能够寻求一种高效可靠的求解方法, 则完全可以用来解决实际生产过程中遇到的难题.

蚁群优化(Ant Colony Optimization, ACO)是一种具有高度创新性的元启发式算法, 最早于 1992 年在 Dorigo 的博士论文中提出. 这种基于种群的元启发式随机搜索算法, 通过模拟真实蚁群协作过程, 在若干蚂蚁相互独立的情况下, 利用信息素参数和一定的最优路径选取机制, 经过若干次迭代从而获取最优解, 实践证明蚁群优化算法可以利用其元启发式、正反馈等特点较好的求解组合优化问题. 而这也是蚁群优化算法研究的起点, 它为后续算法深入改进奠定了基础,

^① 收稿时间:2014-06-13;收到修改稿时间:2014-07-31

文献[1]中, Dorigo 等人针对问题规模变大而引起的蚂蚁系统的收敛性问题提出了具有精英策略(Elitist Strategy)的蚂蚁系统, 而针对蚂蚁系统中的过早收敛问题, Stützle 等人提出了最大最小蚂蚁系统(Max_Min Ant System, MMAS)^[2], 该方法利用信息素限制域防止最优解过早收敛同时取得了较好的结果, 为了进一步克服 AS 中收敛效率和易陷入局部最优的问题, Dorigo 等人又提出了蚁群系统(Ant Colony System, ACS)^[3], 该方法对蚂蚁系统中的路径选择策略做了调整, 并且对信息素更新采取了全局更新和局部更新相结合的策略, 使得人工蚂蚁在开发新路径和利用已有路径之间达到更好的平衡, 文献[4]中, Balaprakash 等人提出了一种基于经验估算的邻域搜索蚁群优化算法, 用来解决旅行商问题(TSP), 在文献[5]中, Vasko, F.J 等人通过大量的统计分析确定了一组最优的关键参数并应用于排序蚁群优化算法中等等。

串行蚁群优化算法在求解大部分优化组合问题时表现出了卓越的求解能力, 但是当问题规模越来越大时, 串行蚁群优化算法很难在短时间内获得理想的解, 所以对传统串行算法进行并行化改进变得极其必要。

利用并行技术优化蚁群算法解质量和效率的思路源于 Dorigo 的工作^[6], Randall 和 Lewis 实现了第一个具有并行计算思想的蚁群优化算法^[7]。在文献[8]中, 扬州大学的陈玲等人提出了为每个进程选择一个与自身进程解的差异性最大的进程进行信息素交换和更新, 同时根据不同的解自适应调整信息交互的时间间隔, 既可以适时的将一个进程上的最优解发送到其他进程并且还可以减少通信时间, 并取得了较高质量的解, 从本质上说, 蚁群优化算法应以并行协同优化计算的方式进行才能真正体现出其天然并行性的优势, 因此, 蚁群优化算法并行化的研究显得尤为重要。

针对上述问题, 提出了一种基于交互的并行蚁群优化算法, 根据一种自定义的并行蚁群信息素的更新策略, 从而增强各个蚂蚁种群的寻优能力, 最后对各个种群最优解分析比较, 得出结果。实验表明, 将本算法用于求解 TSP 问题时间明显缩短, 能够在较短时间内取得较好的加速比和较优的结果。

1 蚁群优化算法的数学模型

自然界中蚂蚁在觅食过程中经过若干条路径从蚁穴到达食物源, 研究发现, 蚂蚁会在它所经过的路径

上留下一种称为信息素的物质, 从而指导后续蚂蚁的移动方向。选择某一路径的蚂蚁越多, 释放的信息素越多, 后续的蚂蚁选择该条路径的概率就会越大。蚂蚁个体之间就是通过这种方式进行交流继而确定到达食物源的最短路径。

接下来以 TSP 问题为例说明基本蚁群优化算法的数学模型:

给定 n 个城市, 旅行商从一个城市出发之后, 分别并且仅访问各个城市一次, 最后回到刚开始出发的城市, 要求找出一条最短的环形路径。

假设蚁群系统中蚂蚁的数量为 m , 在算法初始时刻, 将 m 只蚂蚁随机的分配到 n 个城市中, 蚂蚁 $k(k=1,2,3,\dots,m)$ 在寻优过程中, 根据每条路径上信息素浓度的大小判断下一步的转移方向。蚂蚁 k 当前已经走过的城市使用禁忌表 $tabu_k(k=1,2,3,\dots,m)$ 来存放。蚂蚁在路径搜索过程中, 通过路径上的启发信息以及信息量, 计算状态转移概率, $P_{ij}^k(t)$ 表示在 t 时刻蚂蚁 k 由城市 i 转移到城市 j 的状态转移概率, 其计算公式如公式(1)所示。

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{k \in allowed_k} [\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta} & j \in allowed_k \\ 0 & otherwise \end{cases} \quad (1)$$

其中 $allowed_k = \{ C - tabu_k \}$ 表示的是蚂蚁 k 下一步能够选择的城市场集; α 为信息启发式因子; β 是期望启发因子; $\eta_{ij}(t)$ 是启发函数, 可以表示为 $\eta_{ij}=1/d_{ij}$ 。该公式表示蚂蚁从当前城市 i 转移城市 j 的期望程度。当所有蚂蚁完成一次路径寻优后, 路径 (i,j) 上的信息素可按如下规则进行调整:

$$\tau_{ij}(t+n) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (2)$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t) \quad (3)$$

$$\Delta\tau_{ij}^k(t) = \begin{cases} Q/L_k & (i,j) \in tabu_k \\ 0 & (i,j) \notin tabu_k \end{cases} \quad (4)$$

其中, ρ 表示信息素挥发系数, Q 为常量, L_k 表示蚂蚁 k 本次寻优的路径长度, $\Delta\tau_{ij}(t)$ 表示本次寻优中路径 (i,j) 上的信息素增加量, $\Delta\tau_{ij}^k(t)$ 表示在本次寻优中蚂蚁 k 在路径 (i,j) 上释放的信息素量。

传统蚁群优化算法模型在解决小规模 TSP 问题时能够得到比较理想的结果, 但是对于大规模复杂优化问题, 蚁群优化算法需要较长的搜索时间, 蚁群很难

在较短时间内从大量杂乱无章的路径中找到一条较好的路径. 这主要是因为是在蚁群搜索早期, 各个路径上的信息素相差并不明显. 虽然有正反馈机制, 但是需要经过很长的一段时间, 才能使得较好路径上的信息素浓度明显高于其他路径上的信息素浓度.

另外蚁群优化算法容易出现停滞现象, 出现早熟收敛, 即容易陷入局部最优解. 因为当搜索进行到一定时间后, 所有蚂蚁个体所发现的解几乎一致, 不能进一步得到较好解.

2 邻域搜索技术

邻域搜索算法是解决 TSP 问题的一种改进型算法, 对于求解问题的城市数量一般没有限制, 邻域搜索算法主要通过搜索当前解的邻域来获取改进解, 邻域搜索的原理描述如下: 算法首先在初始解的基础上进行参数调整, 在初始解的邻域中找出一些候选解, 由邻域搜索算法的特定判断策略对候选解与当前解进行不断对比, 优化当前解, 以循环的方式通过不断调整重置邻域搜索的主要参数重新搜索当前解的邻域, 直到满足搜索终止条件.

下面以 2-opt 邻域搜索算法为例, 说明邻域搜索的过程.

当前的较优解. 可以看到较优解中部分城市序列为... $i \rightarrow i+1 \rightarrow j \rightarrow j+1$..., 在这 4 个城市中随机挑选出一个城市, 并在其邻域内搜索更优的解, 假定选择城市 i , 令其邻域的大小为 3, 即在 i 的三个邻域点 $i+1, j, j+1$ 中搜索比当前路径更优的城市序列, 设 d_{ij} 为点 i 和点 j 之间的距离, 其中存在 $d_{ij} + d_{i+1, j+1} < d_{i, i+1} + d_{j, j+1}$, 则用序列 $i \rightarrow j \rightarrow i+1 \rightarrow j+1$ 替换 $i \rightarrow i+1 \rightarrow j \rightarrow j+1$, 替换后序列如图 1(b)所示.

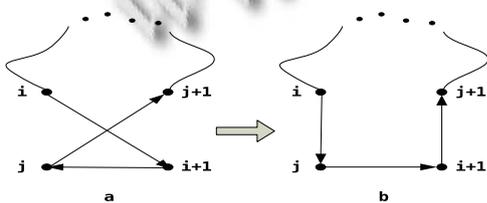


图 1 2-opt 邻域搜索算法演化过程

针对启发式算法来说, 大多数情况下无法得到问题最优解, 只能得到近似最优解, 随着问题规模增大, 由于最优解无法计算, 也很难判断得到的近似解的优

劣, 对于某些问题, 求其整体最优解可能存在不可克服的困难, 但是可以对当前可行解的邻域范围内进行搜索, 寻求得到更优的解, 这就是邻域搜索技术在解决 TSP 问题上存在的优势, 通过对蚁群优化算法得到的可行解进行二次寻优从而改进当前解.

实验过程中发现对蚁群优化算法和邻域搜索算法进行融合改进, 不仅可以发挥蚁群优化算法搜索较优解的能力而且能使邻域搜索算法在解决大规模 TSP 问题时有很好的表现, 从而解决单一的蚁群优化算法易陷入局部最优的难题.

3 基于交互式的并行蚁群优化算法

3.1 并行蚁群优化算法

本文在文献[9]的基础上提出了一种改进的并行交互蚁群策略, 借助 3-opt 邻域搜索算法, 对蚁群系统 (Ant Colony System, ACS)^[10]进行并行化改进, 将一定规模 TSP 问题的求解任务划分为若干并行的子任务并分配给对应的独立蚁群, 各个独立并行蚁群间进行信息交互, 完成全局信息的更新, 当蚁群获取到最优解后, 算法退出. 算法的框架描述如下:

a) 初始化集群环境, 主节点计算城市 $i, j (i, j = 1, 2, 3 \dots n)$ 间的距离矩阵 $distance[i]$, 根据式(5)计算城市 i, j 之间信息素初始值 $pherome[i]$, 计算完成后广播到从计算节点;

$$pherome[i][j] = \begin{cases} trail_0 & i \neq j \\ 0 & i = j \end{cases} \quad (5)$$

令 $\alpha=1.0, \beta=2.0, \rho=0.1, try = 0$;

$trail_0=1.0/n * q, q$ 为定值, 根据问题规模而定;

try 代表蚁群并行求解次数的计数器;

b) 将 m 只蚂蚁随机的分配至集群的 g 个计算节点上, 构成 g 个并行蚁群,

c) 设定 while 循环迭代条件;

d) 开始迭代, g 个蚁群并行构建各自的最优解, 在蚁群 $colony_p (p=1, 2, 3, \dots, g)$ 上, 若蚂蚁 k 经过城市 (i, j) , 则按照公式(6)对路径 (i, j) 进行局部信息素更新, 其中 xi 为常量, 这里取值 0.1;

$$pherome[i][j] = (1 - xi) * pherome[i][j] + xi * trail_0 \quad (6)$$

e) 运用 3-opt 邻域搜索算法对并行蚁群各自的本次迭代最优解进行二次寻优;

f) 每次迭代完成, 各个并行蚁群按照公式(7)和公式(8)分别进行各自的全局信息素更新;

$$pherome[i][j] = (1.0 - \rho) * pherome[i][j] + d_{tau} \quad (7)$$

$$d_{tau} = \rho * 1.0 / (double) length_p \quad (8)$$

其中, $length_p$ 为蚁群 $colony_p(p=1,2,3,\dots,g)$ 迄今最优路径长度.

g) 判断 while 迭代条件是否满足, 若满足, 跳出 while 循环进入 h), 否则返回 d) 继续迭代;

h) 各并行蚁群进行信息素矩阵的规约求和, 主节点广播求和后的信息素矩阵, 各并行蚁群更新信息素矩阵, 完成蚁群间的信息交互;

i) 判别当前是否已收敛, 若算法结果收敛, 则跳出程序, 否则跳入步骤 b), $try++$, 进行下次尝试.

4 实验结果与分析比较

实验过程中, 使用五台同配置并且安装 Ubuntu.11.04.Server 的 PC 作为集群结点, 集群通过一台 H3C 的 SMB-S1224 型千兆以太网交换机实现互相连接.

文中所使用的 TSP 问题数据为海德堡大学研究人员提供的 TSPLIB 实例库^[11]. 本文算法和文献[12]提出的串行 ACS 算法在 TSPLIB 实例库中 10 种问题运行结果如表 1.

表 1 文献[12]算法与本文算法试验结果

TSP 问题	已知最优解	使用算法	运行结果	运行时间
Oliver30	420	文献[12]算法	420	0.011s
		本文算法	420	0.133s
eil51	426	文献[12]算法	426	0.567s
		本文算法	426	0.869s
berlin52	7542	文献[12]算法	7542	0.018s
		本文算法	7542	0.130s
eil76	538	文献[12]算法	538	0.196s
		本文算法	538	0.355s
kroA100	21282	文献[12]算法	21282	0.159s
		本文算法	21282	0.180s
d198	15780	文献[12]算法	15780	28.525s
		本文算法	15780	31.677s
a280	2579	文献[12]算法	2579	1.876s
		本文算法	2579	1.790s
pcb442	50778	文献[12]算法	50778	96.632s
		本文算法	50778	40.512s
pr1002	259045	文献[12]算法	259200	130.513s
		本文算法	259045	41.536s
pr2392	378032	文献[12]算法	378861	291.907s
		本文算法	378541	87.705s

从上述的实验数据可以看出, 当 TSP 问题的城市规模越来越大时, 传统的 ACS 算法表现出了明显的不足, 求解时间过长, 运行结果相对较差.

绘制实验结果的加速比曲线如下图 2 所示:

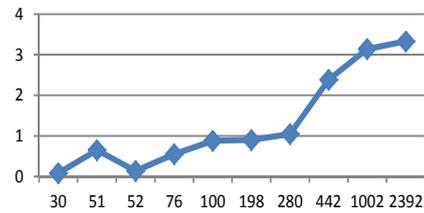


图 2 算法加速比曲线图

本文算法和文献[12]所提出的串行 ACS 算法取得最优解的平均时间以及相应的加速比如下表 2 所示.

表 2 两种算法最优解平均时间以及加速比

TSP 问题	使用算法	取得最优解的平均时间	加速比
Oliver30	文献[12]算法	0.0004s	0.082
	本文算法	0.0032s	
eil51	文献[12]算法	0.1881s	0.652
	本文算法	0.0772s	
berlin52	文献[12]算法	0.0016s	0.138
	本文算法	0.0028s	
eil76	文献[12]算法	0.0192s	0.552
	本文算法	0.0244s	
kroA100	文献[12]算法	0.0156s	0.883
	本文算法	0.0072s	
d198	文献[12]算法	2.8453s	0.900
	本文算法	3.1574s	
a280	文献[12]算法	0.1852s	1.048
	本文算法	0.1664s	
pcb442	文献[12]算法	9.6366s	2.385
	本文算法	4.0354s	
pr1002	文献[12]算法	13.002s	3.142
	本文算法	4.1238s	
pr2392	文献[12]算法	29.0026s	3.328
	本文算法	8.6165s	

分析上述实验数据可以发现, 当 TSP 问题规模较大时, 传统的 ACS 算法并不是求解 TSP 问题的最优解决办法, 上述实验数据说明本文所提出的并行蚁群优化算法比传统串行蚁群优化算法更易获取到理论最优值.

分析表中的实验结果可以看出, 当城市数目较少

时文献[12]算法与本文算法的计算结果都接近已知最优解,当城市数目较大时,本算法明显优于文献[12]中的算法,获得更加近似于理论最优值的解,同时算法运行时间也大大减少。

本算法之所以具有较高的寻优能力是因为它将整个蚁群分解为若干个并行蚁群到对应的计算节点中来加快收敛速度,并能够通过蚁群间的信息交互来避免早熟。

5 结语

本文提出了一种基于交互的并行蚁群优化算法,由于蚁群系统(Ant Colony System, ACS)算法易陷入局部最优的缺陷,通过对传统蚁群系统的并行化改进,实现了并行蚁群间的信息交互更新,改进后的算法不仅能够发挥蚁群优化算法快速寻找较优解的能力,而且在此基础上对较优解进行二次求解以增加获取最优解的机率,同时利用并行蚁群的信息素更新策略能够更快更好的确定最优解,实验证明,该算法在解决大规模旅行商问题上具有明显的优越性,能够在较短时间内获取到较优的解。

参考文献

- 1 Dorigo M, Maniezzo V, Colomi A. The ant system: Optimization by a colony of cooperating agents. *IEEE Trans. on Systems*, 1996, 26(1): 1-13.
- 2 Stützle T, Hoos HH. MAX-MIN ant system. *Future Generation Computer Systems*, 2000, 16(1): 889-914.
- 3 Dorigo M, Gambardella LM. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Trans. on Evolutionary Computation*, 1997, (1): 53-66.
- 4 Balaprakash P, Birattari M, Stützle T. Estimation-based ant colony optimization and local search for the probabilistic traveling salesman problem. *Swarm Intelligence*, 2009, 3(3): 223-242.
- 5 Vasko FJ, Bobeck JD, Governale MA, et al. A statistical analysis of parameter values for the rank-based ant colony optimization algorithm for the traveling salesperson problem. *Journal of the Operational Research Society*, 2011, 62(6): 1169-1176.
- 6 Dorigo M. *Optimization, Learning and Natural Algorithms*[Thesis]. Italy: Politecnico di Milano, 1992.
- 7 Randall M, Lewis A. A parallel implementation of ant colony optimization. *Journal of Parallel and Distributed Computing*, 2002, 62(9): 1421-1432.
- 8 Chen L, Sun HY, Wang S. A parallel ant colony algorithm on massively parallel processors and its convergence analysis for the travelling salesman problem. *Information Science*, 2012, (199): 31-42.
- 9 张志明,安毅生. Linux 机群环境下并行蚁群优化算法的设计与实现[学位论文]. 西安:长安大学, 2012
- 10 Dorigo M, Gambardella LM. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Trans. on Evolutionary Computation*, 1997, (1): 53-66.
- 11 <http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/>.
- 12 <http://iridia.ulb.ac.be/dorigo/ACO/aco-code/public-software.htm>