

基于 MPI 的并行最大最小蚂蚁系统

刘彩云^a, 陈 忠^a, 熊 杰^b

(长江大学 a. 信息与数学学院; b. 电子信息学院, 湖北 荆州 434023)

摘 要: 现有蚁群系统在求解大规模组合优化问题所需的计算时间较长。针对该不足, 提出基于消息传递接口的粗粒度异步协作并行最大最小蚂蚁系统, 在保证解质量的前提下, 降低并行计算中的通信开销。在曙光 4000L 并行机上进行数值实验结果表明, 该系统具有较优的并行加速比和加速效率, 且适合于大规模 TSP 问题的求解。

关键词: 并行最大最小蚂蚁系统; 消息传递接口; 部分异步并行实现; 粗粒度; 多蚁群协作

Parallel Max-Min Ant System Based on MPI

LIU Cai-yun^a, CHEN Zhong^a, XIONG Jie^b

(a. School of Information and Mathematics; b. School of Electronics and Information, Yangtze University, Jingzhou 434023, China)

【Abstract】 Existing Colony system for the large scale optimization combination problem needs long computation time. Aiming at this shortage, this paper presents coarse grained asynchronous parallel implementation Parallel Max-Min Ant System(PMMAS) based on Message Passing Interface(MPI) which can reduce the cost of communication of parallel computation and guarantee the quality of solution. Numerical experiment result which is obtained on the dawn 4000L parallel computer indicates that the system has well speedup and speedup efficiency, and it is suitable for large-scale under the prerequisite of solving Traveling Salesman Problem(TSP).

【Key words】 Parallel Max-Min Ant System(PMMAS); Message Passing Interface(MPI); Partially Asynchronous Parallel Implementation(PAPI); coarse grained; multi-ant colony collaboration

1 概述

蚁群优化算法是一种基于真实蚁群行为的新仿生类算法^[1]。最早由 Dorigo M 等人提出, 并且成功用于求解旅行商问题(Traveling Salesman Problem, TSP)问题、二次分配问题、网络划分问题、蛋白质折叠问题、Web 页面分类问题、路径优化问题^[2]以及其他组合优化问题。蚁群算法能在迭代若干代后, 收敛到几乎最优的解, 但是当求解大规模的组合优化问题时, 和其他优化算法一样存在求解时间过长的问题。并行化是提高求解速度的有效手段之一。消息传递接口(Message Passing Interface, MPI)是目前最为广泛使用的并行计算开发环境, 本文将设计基于 MPI 的并行蚁群优化系统。蚁群优化算法是一个算法家族, 包括基本蚁群算法——蚂蚁系统(Ant System, AS)^[3]及其众多改进算法。到目前为止, 性能最好的是蚁群系统(Ant Colony System, ACS)^[3]和最大最小蚂蚁系统(Max-Min Ant System, MMAS)^[4]。本文选取 MMAS 作为并行最大最小蚂蚁系统(Parallel Max-Min Ant System, PMMAS)的基础。

2 并行最大最小蚂蚁系统

2.1 并行蚁群划分策略

本文给出蚁群划分策略。在并行粒度选择方面, 根据文献[5-8]的研究, 粗粒度的并行策略更有效, 因为这种策略能减少通信时间, 从而获得较优的加速比和加速效率。本文采用粗粒度的并行策略, 即为每个处理器分配多只蚂蚁, 进行并行计算。在蚂蚁协作策略方面, 将同一个处理器上的多只蚂蚁, 看作一个独立的蚁群, 它们拥有完整信息, 包括信息素矩阵、距离矩阵、迭代最优解、 t_{\max} 、 t_{\min} 等。同一个处

理器上的子蚁群, 在本地信息的指导下独立构造解。不同子蚁群间(即不同处理器间), 通过信息交换的方法进行协作。将这种蚁群划分策略称为粗粒度多蚁群策略。

2.2 信息交换频率

根据信息交换频率的不同, 并行算法主要有同步和部分异步 2 种实现模型。文献[5,9]的研究结果表明, 部分异步并行实现模型更有效。因此, 本文采用部分异步并行实现(Partial Asynchronous Parallel Implementation, PAPI)的方式, 让各个蚁群每隔固定的迭代次数进行一次信息交换。各个计算节点在新的信息素矩阵的基础上, 继续开始下一轮计算。采用 PAPI 方式, 能减少网络中传递的信息次数, 从而降低通信时间开销。

2.3 信息交换算法

按 PAPI 蚁群每隔固定迭代次数需进行一次信息交换。常见的信息交换算法是主节点收集各从节点的本地解和本地信息素矩阵, 从中选出得到目前最优解的节点, 再将该节点的本地解和本地信息素矩阵广播给各从节点。对于大规模优化问题, 信息素矩阵巨大(空间复杂度 $O(n^2)$), 常规信息交换方法每次需广播 2 次信息素矩阵, 因此通信时间开销很大。

为提高信息交换效率, 本文设计一种新的信息交换算法。

基金项目: 国家自然科学基金资助项目“基准面旋回理论解析与定量层序地层模型”(40572078); 国家“118”专项基金资助子项目“天然气水合物资源综合评价软件研制与开发”(GZH200200202-3)

作者简介: 刘彩云(1975-), 女, 副教授、硕士, 主研方向: 并行计算; 陈 忠, 教授、博士; 熊 杰, 讲师、硕士

收稿日期: 2010-04-21 **E-mail:** Liucaiyun01@yahoo.com.cn

首先主节点收集各从节点的本地解；然后主节点选出得到目前最优解的从节点，并将该从节点号广播给各从节点；最后由得到目前最优解的从节点将本地解和本地信息素矩阵广播给其他从节点。本信息交换算法每次只需广播一次信息素矩阵，能很好地降低通信时间开销。

2.4 系统实现

下文给出主节点(算法 1)和从节点(算法 2)的并行最大最小蚁群系统。

算法 1(主节点算法)

步骤 1 读取用户参数 β γ ρ $seed$ ，初始化信息素矩阵和距离矩阵；

步骤 2 广播 β γ ρ $seed$ ，信息素矩阵、距离矩阵和城市数 N 到各节点；

步骤 3 如果不满足算法结束条件，则反复执行步骤 4~步骤 9，否则退出算法；

步骤 4 按照 2.1 节所述粗粒度多蚁群策略，划分蚁群，并随机平均分配蚂蚁到各个城市；

步骤 5 按照 MMAS，本地迭代 N 次求解；

步骤 6 收集各个节点的求解结果；

步骤 7 根据 2.3 节信息交换算法，与各从节点交换信息；

步骤 8 收集各个节点的结束标志，计算全局结束标志，并广播给各个节点；

步骤 9 转步骤 3，开始下一次循环。

算法 2(从节点算法)

步骤 1 接收主节点广播的 β γ ρ $seed$ ，信息素矩阵、距离矩阵和城市数 N ；

步骤 2 如果不满足算法结束条件，则反复执行步骤 3~步骤 7，否则退出算法；

步骤 3 随机平均分配蚂蚁到各个城市；

步骤 4 按照 MMAS，本地迭代 N 代求解；

步骤 5 根据 2.3 节信息交换算法，与主节点交换信息；

步骤 6 发送本地结束标志给主节点，并接收全局结束标志；

步骤 7 转步骤 2，开始下一次循环。

3 数值实验结果与分析

硬件平台是曙光天潮 4000L 并行机，软件环境是 Linux+MPI+C。利用 MPI 函数库，编写 C 语言程序，实现本文提出的并行最大最小蚁群系统。通过实验研究实例规模、计算节点数对本文设计的并行最大最小蚁群系统性能的影响。

3.1 评价指标与评价方案

本文主要从 2 个方面考察并行最大最小蚁群系统：

(1)是否适用于所有规模的 TSP 问题的求解；

(2)在不同数量的计算节点并行计算时，是否同样高效。

本文设计如下 2 个评价方案：

(1)固定采用 2 个节点并行计算，分别以不同规模的经典 TSP 实例，运行串行 MMAS(Serial MMAS, SMMAS)和 PMMAS，测试 PMMAS 的并行性能；

(2)以同一个经典 TSP 为例，分别采用 2 个、4 个、8 个、16 个节点进行并行计算，测试 PMMAS 的在不同计算节点并行计算时的性能。

3.2 参数设置

本文将 PMMAS 用于求解各种经典 TSP 实例^[10]，主要参数设置为 $\alpha=1$ 、 $\beta=2$ 、 $\rho=0.2$ 、 $m=25$ 、 $t_0=1/\rho C^{bs}$ ，其中， α 、 β 分别为信息素影响因子和启发信息影响因子； ρ 为信息素的

蒸发率； m 、 n 分别为蚂蚁数和城市数； t_0 为信息素初始值； C^{bs} 表示由最近邻启发式方法构造的路径长度。

信息素大小被限制在 $t_{\max}=1/\rho C^{bs}$ 到 $t_{\min}=t_{\max}(1-\sqrt[n]{0.05})/((avg-1)\sqrt[n]{0.05})$ 之间，其中， avg 是每只蚂蚁在构建解的每一步中面临的不同选择的平均个数。

3.3 实验结果

(1)实例规模对并行性能的影响

利用 2 个节点对不同规模的 TSP 实例进行求解，为了在同等计算条件下对比 SMMAS 与 PMMAS 的性能，实验中将 2 种算法的最大执行时间设为相同。现将 PMMAS、SMMAS 算法对不同实例重复运行 20 次，实验结果见表 1 所示。其中， t_{\max} 表示最大运行执行时间； t_{avg} 为求得的最优解平均计算时间； i_{avg} 为平均迭代次数。

表 1 SMMAS、PMMAS 求解 TSP 的结果比较

实例	算法	平均值	t_{\max}/s	t_{avg}/s	i_{avg}	平均误差/(%)
D198	PMMAS	15 781.1	20	6.34	370.70	0.000 07
	SMMAS	15 780.0	20	5.53	574.00	0.000 00
Gr666	PMMAS	294 546.0	100	42.53	762.80	0.063 90
	SMMAS	294 422.1	100	56.95	1 647.35	0.021 80
Pcb1173	PMMAS	56 906.0	200	87.26	895.80	0.024 60
	SMMAS	56 909.4	200	126.89	1 895.25	0.030 60
fl1577	PMMAS	22 320.4	400	140.89	1 107.20	0.320 90
	SMMAS	22 323.1	400	234.14	3 288.55	0.333 00
pr2392	PMMAS	378 821.3	1 000	421.65	1 661.85	0.208 80
	SMMAS	378 823.0	1 000	750.57	4 359.10	0.209 20
rl5915	PMMAS	569 388.6	2 000	957.83	705.50	0.682 30
	SMMAS	569 668.6	2 000	1 733.94	2 082.90	0.731 80

(2)计算节点数对并行性能的影响

以 fl1577 实例为例，分别采用 2 个、4 个、8 个、16 个节点进行计算，重复运行 20 次，实验结果见表 2。

表 2 采用不同计算节点数求解 TSP 的结果比较

实例	节点数	t_{avg}/s	平均误差/(%)	加速比	加速效率
fl1577	1	234.14	0.333	1.000 0	1.000 0
	2	140.89	0.333	1.661 9	0.830 9
	4	102.4	0.325	2.286 5	0.571 6
	8	86.63	0.318	2.702 8	0.337 8
	16	79.38	0.346	2.949 6	0.184 4

3.4 结果分析

本文利用表 2，计算出加速比、加速效率，得到加速比与实例规模的关系(见图 1)、加速效率与实例规模的关系(见图 2)。由图 1、图 2 可以看出，对于固定 2 个节点参与计算，加速比、加速效率均随实例规模的增长逐渐提高。对于大型实例规模的求解(如 rl5915)，加速比接近于上限 2，加速效率接近于上限 1。结果表明本文所设计的算法更适合于大规模的 TSP 问题的求解。

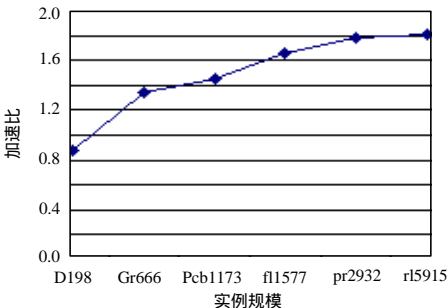


图 1 加速比与实例规模的关系

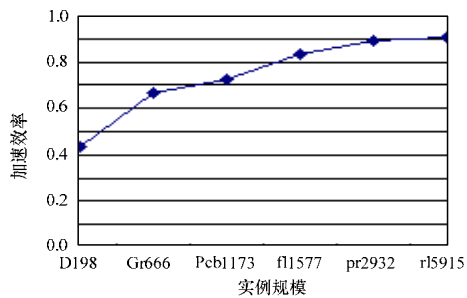


图2 加速效率与实例规模的关系

根据表3, 得到加速比与计算节点数的关系(见图3), 加速效率与计算节点数的关系(见图4)。

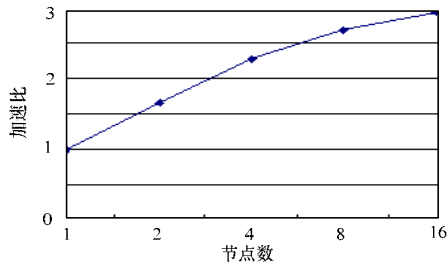


图3 加速比与计算节点数的关系

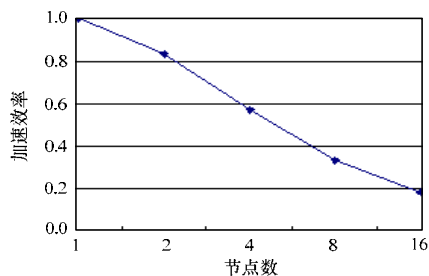


图4 加速效率与计算节点数的关系

由图3、图4可以看到, 随着计算节点数的增加, 加速比单调上升, 但增长速度逐渐减慢; 加速效率单调下降, 但下降速度逐渐减慢。可见, 更多节点参与计算, 虽然能提高求解速度, 获得较好的加速比, 但加速效率却不理想。通过分析可知, 这是因为更多节点参与计算, 虽然能带来更强的计算能力, 但也会导致通信开销逐步增大。因此, 不断的增加计算节点, 并不一定总能获得理想的加速效果, 需要综合考虑加速比和加速效率2个因素, 以获得满意的加速效果。

4 结束语

本文采用并行蚁群划分策略、信息交换频率和信息交换算法, 提出一种新的并行最大最小蚂蚁系统。通过数值实验研究了实例规模、计算节点数对并行性能的影响。实验结果表明, 该系统在适当的并行规模(计算节点数)下求解 TSP 问题, 能显著加快求解速度。

参考文献

- [1] Colomi A, Dorigo M, Maniezzo V. Distributed Optimization by Ant Colonies[C]//Proc. of European Conference on Artificial Life. Paris, France: [s. n.], 1991.
- [2] 王 罡, 冯艳君. 基于蚁群优化算法的旋转货架拣选路径规划[J]. 计算机工程, 2010, 36(3): 221-223.
- [3] Dorigo M, Grambardella L M. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem[J]. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 53-56.
- [4] Stutzle T, Hoos H. MAX-MIN Ant System and Local Search for the Traveling Salesman Problem[C]//Proc. of International Conference on Evolutionary Computation. Indianapolis, USA: IEEE Press, 1997.
- [5] Bullnheimer B, Kotsis G, Strauss C. Parallelization Strategies for the Ant System[C]//Proc. of Conference on High Performance Software for Nonlinear Optimization: Status and Perspectives. Ischia, Italy: [s. n.], 1997.
- [6] Stutzle T. Parallelization Strategies for Ant Colony Optimization[C]//Proc. of the 5th International Conference on Parallel Problem Solving From Nature. London, UK: Springer-Verlag, 1998.
- [7] Manfrin M. Parallel Ant Colony Optimization for the Traveling Salesman Problem[C]//Proc. of ANTS'06. Brussels, Belgium: [s. n.], 2006.
- [8] Benkner S. Communication Strategies for Parallel Cooperative Ant Colony Optimization on Cluster and Grids[C]//Proc. of PARA'04 Workshop on State-of-the-art in Scientific Computing. Lyngby, Denmark: [s. n.], 2004.
- [9] Middendorf M, Reischel F, Schmeck H. Multi Colony Ant Algorithm[J]. Journal of Heuristics, 2002, 8(3): 305-320.
- [10] Ruprecht-Karls-Universitat-Heidelberg. TSPLIB[EB/OL]. [2009-11-22]. <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95>.

编辑 陆燕菲

(上接第199页)

参考文献

- [1] Loy G, Barnes N. Fast Shape-based Road Sign Detection for a Driver Assistance System[C]//Proc. of IEEE International Conference on Intelligent Robots and Systems. Nice, France: IEEE Press, 2004: 70-75.
- [2] Cyganek B L. Road-signs Recognition System for Intelligent Vehicles[C]//Proc. of the 2nd International Workshop on Robot Vision. Auckland, New Zealand: [s. n.], 2008: 219-233.
- [3] Jun-Taek O. Segmentation and Recognition of Traffic Signs Using Shape Information[C]//Proc. of ISVC'05. Lake Tahoe, Nevada, USA: [s. n.], 2005: 519-526.

- [4] 黎海兵, 易卫东. 一种高效检测图像中是否有三角形的算法[J]. 中国图象图形学报, 2008, 13(6): 456-460.
- [5] Loy G, Zelinsky A. Fast Radial Symmetry for Detecting Points of Interest[J]. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2003, 25(8): 959-973.
- [6] Duda R, Hart P. Use of the Hough Transform to Detect Lines and Curves in Pictures[J]. Communications of the ACM, 1972, 15(1): 11-15.
- [7] Bascon S M. Road-sign Detection and Recognition Based on Support Vector Machines[J]. IEEE Trans. on Intelligent Transportation Systems, 2007, 8(2): 264-278.

编辑 金胡考

