多构造蚁群优化求解置换流水车间调度问题

刘延风 刘三阳
(西安电子科技大学应用数学系 西安 710071)

摘 要 针对置换流水车间调度问题,提出了一种多构造蚁群优化求解算法。在该算法中,蚁群采用两种方式构造解,分别是基于NEH (Nawaz-Enscore-Ham,NEH)启发式算法和Rajendran启发式算法,并根据解的质量,自适应地调整两种构造方式在蚁群中所占的比例。对置换流水车间调度问题的基准问题测试表明,提出的算法是有效的。

关键词 多构造蚁群优化,置换流水车间调度,NEH启发式算法,Rajendran启发式算法

中图法分类号 TP38 **文献标识码** A

Multi-construction Ant Colony Optimization Algorithm for Permutation Flow Shop Scheduling

LIU Yan-feng LIU San-yang
(Department of Applied Mathematics, Xidian University, Xi'an 710071, China)

Abstract A Multi Construction Ant Colony Optimization Algorithm for Permutation Flow Shop Scheduling was proposed. In this algorithm, solutions are constructed through two modes, which are based on Nawaz-Enscore-Ham heuristics and Rajendran heuristics respectively. Then the proportion of construction modes is adjusted adaptively according to quality of solution constructed. Simulation results and comparisons based on benchmarks demonstrate the effectiveness of the algorithm.

Keywords *Permutation flow shop scheduling, Multi construction ant colony optimization*, NEH heuristics, Rajendran heuristics

1 引言

置换流水车间调度问题是许多实际流水线生产调度的简化模型。该问题描述如下:n个工件需要在m台机器上加工,所有工件以相同的顺序在每一台机器上完成加工,每台机器每次最多只能加工一个工件,每个工件每次只能由一台机器加工。已知每个工件在每台机器上的加工时间,求出工件加工顺序,使得按照该加工顺序加工工件对应的总完工时间最小。研究表明,该问题属于NP难题[1],求解非常困难。因此,开发和研究高效的优化算法,具有非常重要的理论和实际意义。

置换流水车间调度的求解方法通常分为3种类型:精确方法、启发式算法以及现代优化算法。精确方法主要有分枝定界法、动态规划法等。由于是NP(Non-deterministic Polynomial-time,NP)难题,精确算法仅仅适用于小规模问题的求解。启发式算法比较多,例如Gupta法、Palmer法、CDS(Campleu-Dudek-Smith,CDS)法、Rajendran法、NEH法等。这些算法虽然能够快速构造解,但是通常解的质量较差。值得指出的是,在启发式算法中,NEH法和Rajendran法性能最好。现代优化算法主要包括遗传算法(Genetic Algorithm,GA)[2]、禁忌搜索法[3]、模拟退火法(Simulated Annealing,SA)[4]以及蚁群系统优化算法(Ant Colony System,ACS)[5]、蚁群优化算法(Ant Colony Optimization,ACO)[6]等。

在蚁群优化算法求解置换流水车间调度问题中,解的构造有的就没有利用启发式信息,有的虽然利用了启发式信息但是仅仅利用了单一启发式信息,例如文献[5]中解的构造是基于Palmer启发式算法的。可是,如前所述,置换流水车间调度问题有多种启发式构造算法,这些启发式算法大多可以为蚁群算法所采用。受此启发,考虑到NEH和Rajendran启发式算法性能最好,作者构造了一种多构造型蚁群算法求解置换流水车间调度问题。在该算法中,蚁群采用两种方式构造解,分别是基于NEH启发式算法和Rajendran启发式算法。然后,根据解的质量,自适应地调整两种构造方式在蚁群中所占的比例。通过对Carl－Car8以及Taillard系列的基准测试问题的实验证明了所提算法的有效性。

2 蚁群优化

蚁群算法是由意大利学者Dorigo,Maniezzo和Colorni等人提出的一种新型分布式现代启发式算法[7]。受蚂蚁寻找食物行为的启发,模仿蚁群的行为求解组合优化问题,是蚁群算法的本质。蚂蚁在寻找食物时,在它走过的路径上释放一种叫做信息素的物质,其他蚂蚁能够探测出这种物质并受其影响。若某些路径上走过的蚂蚁越多,留在该路径上的信息素也就越多,后来的蚂蚁选择该路径的概率也越高。蚂蚁正

到稿日期:2009-02-09 返修日期:2009-05-13

刘延风(1970－),男,讲师,主要研究方向为智能优化算法及其应用,E-mail:teacher2003@tom.com;刘三阳(1959－),男,教授,主要研究方向为最优化理论与算法。

是通过信息素作为媒介，最终走上一条它们并未意识到的连接蚁巢和食物之间的最短路径上。下面以旅行商问题(Traveling Salesman Problem，TSP)为例简单介绍蚁群算法。

TSP 问题：已知 n 个城市之间的距离，寻找一条访问每一个城市且仅访问一次的最短长度环游。

设 d_{ij} 是城市 i 到城市 j 之间的路径长度，τ_{ij} 表示路径 ij 上的信息素，η_{ij} 表示路径 ij 的能见度，$\eta_{ij}=1/d_{ij}$。第 k 只蚂蚁从城市 i 到城市 j 的转移概率为

$$p_{ij}^{k}=\begin{cases}\dfrac{[\tau_{ij}]^{\alpha}\cdot[\eta_{ij}]^{\beta}}{\sum\limits_{s\in allowed_k}[\tau_{is}]^{\alpha}\cdot[\eta_{is}]^{\beta}}, & j\in allowed_k\\ 0, & \text{其他}\end{cases}\tag{1}$$

其中，$allowed_k$ 表示第 k 只蚂蚁未访问的城市。

蚁群算法的流程[8]如下。

Step1　初始化。设置迭代次数 $counter=0$、最大迭代次数为 $Nc\ \max$，为信息素 τ_{ij} 赋初始值 $\tau_0=1/(n\cdot L_m)$，其中 L_m 表示按照最近邻启发式算法得到的环游长度。

Step2　将 m 只蚂蚁放在不同的城市上。

Step3　每一只蚂蚁按照式(1)选择下一个城市。

Step4　所有的蚂蚁得到各自的环游以后，对得到的环游做局部寻优。

Step5　按照下面的公式更新信息素。

$$\tau_{ij}=(1-\rho)\cdot\tau_{ij}+\Delta\tau_{ij}$$

其中，$0<\rho\leqslant 1$ 为挥发系数；

$$\Delta\tau_{ij}=\sum_{k=1}^{m}\Delta\tau_{ij}^{k}$$

其中，$\Delta\tau_{ij}^{k}$ 为第 k 只蚂蚁在路径 ij 留下的信息素，且

$$\Delta\tau_{ij}^{k}=\begin{cases}\dfrac{Q}{L_k}, & \text{第 } k \text{ 只蚂蚁经过路径 } ij\\ 0, & \text{否则}\end{cases}$$

其中，Q 为常数，L_k 为第 k 只蚂蚁得到的环游长度。

Step6　$counter++$。如果 $counter<Nc\ \max$，返回 step2；否则结束。

3　多构造蚁群优化求解置换流水车间调度问题

3.1　Rajendran 启发式算法

Rajendran 启发式算法[9]步骤如下：

Step1　对每一个工件 i 计算参量 $T_{i1}=\sum\limits_{j=1}^{m}(m-j+1)t_{ij}$，并按照递增顺序进行排列。

Step2　选取队列中的第一个工件生成部分解，然后选取队列中的第 k 个工件尝试插入到已经生成部分解的第 l 位置，其中 $[k/2]\leqslant l\leqslant k$，$k\geqslant 2$。

Step3　令其中使得部分最大流程时间最小的部分解为新的部分解。

Step4　按照上述方法依次将队列中剩余的工件插入，从而得到一个解。

3.2　NEH 启发式算法

NEH 启发式算法[10]是由 Nawaz，Enscore 和 Ham 共同提出的，该算法的基本思想是赋予总加工时间越长、体积越大的工件在排列中的插入优先权。算法通过每一步加入一个新工件，求得最好的局部解，最后构造出工件的加工顺序。

NEH 算法步骤如下。

Step1　按工件在机器上的总加工时间递减的顺序排列 n 个工件。

Step2　取前两个工件调度，使部分最大流程时间达到极小。

Step3　从 $k=3$ 到 n 把第 k 个工件插入到 k 个可能的位置，求得最小部分的最大流程时间。

3.3　多构造蚁群优化算法

多构造蚁群优化算法包括两种构造方式：第一种是基于 Rajendran 法，第二种是基于 NEH 法。

为了利用多构造蚁群优化算法求解置换流水车间调度问题，首先计算出每个工件的总加工时间 S_i 和参量 T_{i1} $(i=1,2,\cdots,n)$，按照 NEH 启发式算法得到最大流程时间 $makespan$。

其次重新定义信息素和能见度，使得能见度和信息素能够引导蚂蚁构造好解。τ_{ij} 定义为工件 i 排在第 j 位置的信息素，其初始值 $\tau_0=1.0/(n*makespan)$。这样定义信息素使得构成好解的工件及位置能够以较大的概率被选中。在基于 NEH 法的构造中，能见度定义为 $\eta_i=S_i$；在基于 Rajendran 法的构造中，能见度定义为 $\eta_i=T_{i1}$。这样定义能见度使得蚁群算法可以利用性能良好的 NEH 启发式信息和 Rajendran 启发式信息构造出好解。

再次，第 k 只蚂蚁第 j 步选择工件 i 的概率为

$$p_{ij}^{k}=\begin{cases}\dfrac{[\tau_{ij}]^{\alpha}[\eta_i]^{\beta}}{\sum\limits_{u\in allowed_k}[\tau_{uj}]^{\alpha}[\eta_u]^{\beta}}, & \text{工件 } i \text{ 未被蚂蚁 } k \text{ 选中}\\ 0, & \text{其他}\end{cases}\tag{2}$$

其中，$allowed_k$ 表示从未被第 k 只蚂蚁选中过的工件集合。

第四，对蚁群构造的部分解做局部寻优。如果蚁群的构造方式为基于 Rajendran 法的，局部优化采用 Rajendran 法中的 Step2 和 Step3 所述方法；如果蚁群的构造方式是基于 NEH 法的，局部寻优采用 NEH 启发式算法中的 Step2 和 Step3 所述方法。

第五，对上面构造的解做插入型局部寻优。具体方法为将排在第 $i(i=0\sim n-2)$ 个位置的工件依次插入到第 $i+1\sim n-1$ 位置。

第六，分配两种构造方式的比例。除了第一次是按照平均方式分配以外，两种构造方式在下一次循环中所占的比例均是根据各构造类型对应的平均最大流程时间自适应调整。

第 k 只蚂蚁选择第 1 种构造类型的概率为

$$p_1^{k}=\frac{atd_2}{atd_1+atd_2}\tag{3}$$

第 k 只蚂蚁选择第 2 种构造类型的概率为

$$p_2^{k}=\frac{atd_1}{atd_1+atd_2}\tag{4}$$

其中，atd_1 和 atd_2 分别为第 1 种和第 2 种构造类型的平均最大流程时间。

多构造型蚁群算法的流程如下。

Step1　初始化。设置迭代次数 $counter=0$，设置最大迭代次数为 $Nc\ \max$，为信息素 τ_{ij} 赋初始值 $\tau_0=1.0/(n*makespan)$，其中 $makespan$ 为按照 NEH 启发式算法得到的最大流程时间。

Step2　将 na 只蚂蚁平均分配给两种构造类型。

Step3　基于 NEH 构造类型的蚂蚁能见度定义为 $\eta_i=S_i$，按照式(2)构造解；基于 Rajendran 构造类型的蚂蚁能见

度定义为 $\eta_i = T_{ij}$。

Step4　每一只蚂蚁都按照式(2)选择工件。

Step5　对蚁群构造的部分解根据其构造类型分别按照各自启发式算法中 Step2 和 Step3 所述方法做局部优化。

Step6　构造所有蚂蚁解完毕后，对得到的解做插入型局部寻优。

Step7　为下一次循环的构造类型分配蚂蚁个数。首先，每一个构造类型各分配一只蚂蚁。然后，计算各构造类型对应的平均最大流程时间。最后，按照式(3)、式(4)分配余下的蚂蚁个数。

Step8　按照蚁群算法中的方法更新信息素。

Step9　*counter*++。

如果 *counter*<*Nc* max，返回 Step3；否则，结束。

4　实验结果

为了验证算法的有效性，首先选择了文献[11]的 Car 系列基准测试问题进行测试。

硬件环境的处理器主频为 1.70GHz，内存 504M。操作系统为 Windows XP，算法用 VC++6.0 编程实现。参数设置为蚂蚁个数 $na=10$，挥发系数 $\rho=0.1$，$\alpha=1.0$，$\beta=4.0$，最大迭代次数为 30。每个算例随机运行 20 次，用偏离最优值的平均相对误差作为解的性能。得到的结果与 NEH 启发式算法、文献[12]提出的改进遗传算法(Improved Genetic Algorithm，IGA)以及文献[13]提出的量子遗传算法(Quantum inspired Genetic Algorithm，QGA)进行比较，结果如表 1 所列。

表 1　Car 系列基准测试集不同算法性能结果比较

测试问题	最优值	本文算法	NEH	IGA	QGA
Car1	7038	0	0	0	0
Car2	7166	0	2.93	2.10	2.10
Car3	7312	0	1.19	1.31	1.31
Car4	8003	0	1.57	0	0
Car5	7720	0	3.83	0.98	0.98
Car6	8505	0	3.15	0.80	0.80
Car7	6590	0	0	0.16	0.16
Car8	8366	0	2.37	0.47	0.47

从表 1 可以看到，本文提出的算法在 30 次迭代中，找到了 Car 系列基准测试集的全部最优解，明显优于表中的 NEH 启发式算法以及改进遗传算法和量子遗传算法。

为了进一步验证算法有效性，采用 ORLIB[14] 中的 Taillard 系列问题中的部分实例进行测试，其中机器数 m 为 5，10，20，工件数 n 为 20，50，100，这样的 m/n 组合共有 8 组，每一组组合下又有 10 个实例。对于每种 m/n 组合下的实例分别计算 5 次，取 5 次中的最好解与实例上界比较得到相对误差，得出这 10 个实例与上界的平均相对误差作为解的性能指标。得到的结果比较如表 2 所列。参数设置为最大迭代次数为 100，蚂蚁个数 $na=20$，其他设置不变。

表 2　与其他算法性能比较

问题规模(m/n)	本文算法性能	ACS 性能	SA 性能	GA 性能
5/20	0	1.19	1.27	1.61
5/50	−1.17	0.43	0.78	0.45
5/100	−0.04	0.22	0.56	0.23
10/20	0.09	1.70	1.71	2.29
10/50	0.82	1.89	1.98	2.28
10/100	0.49	1.22	1.33	1.25
20/20	0.07	1.60	0.86	1.95
20/50	2.19	2.71	2.86	3.44

从表 2 可以看出，本文所提算法的性能明显好于文献[5]提出的 ACS 法以及文献[5]所引用的模拟退火法(SA)和遗传算法(GA)。

为了比较多构造蚁群优化和单构造蚁群优化算法的性能，采用了上述测试对象、参数设置以及性能指标。分别运行基于 NEH 的蚁群优化算法和基于 Rajendran 的蚁群优化算法，得到的性能比较结果如表 3 所列。

表 3　多构造和单一构造蚁群优化的性能比较

问题规模 (m/n)	本文算法性能	基于 NEH 蚁群算法性能	基于 Rajendran 蚁群算法性能
5/20	0	0.04	0
5/50	−1.17	0.01	0.02
5/100	−0.04	−0.04	0
10/20	0.09	0.17	0.15
10/50	0.82	1.01	1.54
10/100	0.49	0.57	0.90
20/20	0.07	0.08	0.61
20/50	2.19	2.25	2.96

从表 3 可见，多构造蚁群优化的性能明显好于基于 NEH 的和基于 Rajendran 的单一构造蚁群优化。这说明多构造策略通过增加解的多样性提高了算法的性能。

结束语　本文提出了一种多构造型蚁群优化算法来求解置换流水车间调度问题。该算法将 NEH 启发式算法和 Rajendran 启发式算法有机地结合起来。通过对 Car 系列和 Taillard 系列的基准问题测试表明，提出的算法是有效的。

参 考 文 献

[1] Garey E L, Johnson D S, Sethi R. The complexity of flow-shop and job-shop scheduling[J]. Mathematics of Operations Research, 1976, 1: 117-129

[2] 王自强，冯博琴. 车间流程的免疫调度算法[J]. 西安交通大学学报，2004，38：1031-1034

[3] Daya M B, Fawsan M A. A tabu search approach for the flow shop scheduling problem[J]. European Journal of Operational Research, 1996, 91: 160-175

[4] Low C, Yeh J Y, et al. A robust simulated annealing heuristic for flow shop scheduling problems[J]. International Journal of Advanced Manufacturing Technology, 2004, 13: 762-767

[5] Ying Kuo-Ching, Liao Ching-Jong. An ant colony system for permutation flow-shop sequencing[J]. Computers & Operation Research, 2004, 31: 791-801

[6] 王笑蓉，吴铁军. Flowshop 问题的蚁群优化调度方法[J]. 系统工程理论与实践，2003，23：65-71

[7] Colorni A, Doigo M. Heuristics from nature for hard combinatorial optimization problems[J]. International Trans Operational Research, 1996, 3: 1-21

[8] Dorigo M, Stutzle T. Ant Colony Optimization[M]. Cambridge: MIT Press, 2003

[9] 王凌. 车间调度及其遗传算法[M]. 北京：清华大学出版社，2003

[10] Nawaz M, Enscore E, Ham I. A heuristic algorithm for the m-machine n-job flow shop sequencing problem[J]. Omega, 1983, 11: 11-95

[11] 王凌. 智能优化算法及其应用[M]. 北京：清华大学出版社，2003

[12] Lyer S K, Saxena B. Improved genetic algorithm for the permutation flowshop scheduling problem[J]. Computers& Operations Research, 2004, 31: 593-606

[13] Wang Ling, Wu Hao, Zheng Da-zhong. A quantum-inspired genetic algorithm for flowshop scheduling[J]. Lecture notes in Computer Science, 2005, 3612: 417-423

[14] http://people.brunel.ac.uk/~mastjjb/jeb/info.html