

# 求解 Job Shop 调度问题的混合蚁群算法研究

宋晓宇, 王 丹

(沈阳建筑大学信息与控制工程学院, 沈阳 110168)

**摘 要:**为了解决单一算法求解 Job Shop 调度问题存在的不足, 该文提出了一种混合算法, 将蚁群算法用于全局搜索。针对蚁群算法易于陷入局部最优的情况, 提出了一种基于关键工序的邻域搜索方法, 将使用此邻域搜索方法的 TS 算法作为局部搜索策略。利用 TS 算法较强的局部搜索能力, 提高了蚁群算法的优化能力, 达到改善 Job Shop 调度问题解的质量。实验结果表明, 混合算法在较短的时间内, 找到了 FT10、LA24、LA36 等典型 benchmarks 问题的最优解, 得到的 makespan 的平均值较并行遗传算法(PGA)和 TSAB 算法均有所提高。

**关键词:** 蚁群算法; 禁忌搜索; 混合算法; Job Shop 调度

## Research on Hybrid Ant Colony Algorithm for Job Shop Schedule

SONG Xiaoyu, WANG Dan

(School of Information & Control Engineering, Shenyang Jianzhu University, Shenyang 110168)

**【Abstract】** It is hard to get solutions in job shop schedule problem using by single algorithms. This paper proposes a hybrid algorithm for job shop schedule. The ant colony algorithm which plunges the local situation easily is used as a global search algorithm. In addition it proposes taboo search algorithm based on a new neighborhood search method, and the TS algorithms in the method are used as local search algorithm. Because the TS algorithms have the stronger local search ability, and it can overcome the disadvantages of ant colony algorithms, so it gets satisfied solutions for job shop scheduling. The experimental results show that this algorithm can solve typical benchmarks problems efficiently, such as FT10, LA24, LA36 efficiently. The hybrid algorithm gets higher the average agreement compared with TSAB and PGA.

**【Key words】** Ant colony algorithm; Taboo search; Hybrid algorithm; Job shop schedule

Job Shop 调度问题是著名的优化组合问题, 广泛存在于生产实际中, 是调度问题中的典型难题。许多学者对此进行了研究, 推动了相关算法的发展, 并为其他领域类似问题的解决提供条件和手段。蚁群算法是一种应用于组合优化问题的启发式搜索算法, 它是由意大利学者 M. Dorigo<sup>[1,2]</sup> 等人首先提出来的。Stutzle 和 Hoos 在 2000 年提出 MMAS<sup>[3]</sup>, 以此来改进蚁群算法。本文针对 Job Shop 调度问题, 采用 MMAS 算法进行全局搜索, 并使用改进的 G&T(Giffler&Thompson) 算法构造 MMAS 的解。根据问题的特征, 提出一种基于关键工序的邻域搜索策略, 并将使用此邻域搜索策略的禁忌搜索算法(TS)作为局部搜索策略。将 2 种搜索策略构造混合算法, 以提高算法的优化性能, 达到改善 Job Shop 调度问题解的质量。实验结果表明, 这种混合搜索策略是有效的, 在优化效果上, 该方法优于并行遗传算法和禁忌搜索算法。

### 1 模型求解

#### 1.1 算法的混合策略

Colomi 等首次使用蚁群算法解决 Job Shop 调度问题<sup>[4]</sup>, 但其求解效果一般<sup>[5]</sup>。MMAS 等蚁群算法均存在的以下弊端: (1) 当蚁群算法陷于停顿, 表明某些不在最优解上的信息素得到有效增强, 正反馈机制使得算法无法跳出局部状态; (2) 与其他优化算法相比, 蚁群算法搜索时间过长。特别是当初始信息匮乏时, 需要大量的搜索时间。禁忌搜索算法<sup>[6]</sup>(TS)是迄今为止解决 Job Shop 问题较好的算法之一, TS 的计算速度比较快; 同时, 研究结果表明, TS 在整个搜索域内, 能提供比其他算法更好的引导。

本混合算法策略主要是对 G&T(Giffler&Thompson) 算法进行改进, 用于构造蚁群算法的解, 进行全局搜索。根据 Job Shop 调度问题解的特征, 提出一种基于关键工序的邻域搜索方法(见 1.3 节), 并将此邻域搜索方法的禁忌搜索算法(TS)作为局部搜索策略, 构造混合优化算法(TSANT), 当蚁群算法产生新的种群之后, 以概率  $P_i$  选取种群中的个体, 利用 TS 算法对其进行局部优化, 将优化的结果作为蚁群算法本代的最终的种群, 然后再执行蚁群算法的其余步骤, 实现混合算法求解。当蚁群算法进化若干代以后, 计算收敛因子  $cf$ , 当  $cf$  的值大于 0.99, 则对算法进行软复位, 即对信息素的值重置初始值, 重新进行搜索, 解决算法陷入停滞的问题。 $cf$  的计算公式如下:

$$cf = 2 \cdot \left( \frac{\sum_{ij \in \tau} \max\{\tau_{ij} - \tau_{min}, \tau_{ij} - \tau_{max}\}}{|\tau| \cdot (\tau_{max} - \tau_{min})} \right) - 0.5 \quad (1)$$

具体算法(TSANT)如下:

Input: 一个 Job Shop 调度问题 P

Output: 算法搜索到的最优解  $S_{bs}$

Begin  $S_{bs} \leftarrow \text{NULL}; S_{rb} \leftarrow \text{NULL}; S_{ib} \leftarrow \text{NULL};$

//  $S_{ib}$ : 每一次迭代的最优解

iter = 0;  $cf \leftarrow 0; P_i \leftarrow 0.2; // S_{rb}$ : 算法在软复位后的最优解。

**基金项目:** 国家“973”计划基金资助重点项目(2002CB312200); 国家自然科学基金资助重点项目(70431003)

**作者简介:** 宋晓宇(1963-), 男, 教授, 主研方向: 组合优化, 图像处理, MIS; 王 丹, 硕士生

**收稿日期:** 2006-02-27 **E-mail:** xysong@sina.com

```

n ← DetermineNumberOfAnts (P)//初始化种群大小
InitializePheromoneValues(τ) //初始化信息素表
while 未满足终止条件 do
for j = 1 to n do
ConstructionSolution () //构造解
end do
ApplyLocalSearch() //局部搜索
ParallelTabooSearch(Pi) //并行TS搜索
Sib ← argmin{Cmax(S)}
Update(Sib, Sbs, Srb)
ApplyPheromoneUpdate(τ, Sib) //更新信息素
cf ← ComputeConvergenceFactor(τ) //计算收敛因子
if cf > 0.99 then
ResetPheromoneValues(τ) //重置信息素
Srb ← NULL
end if
iter++
end while
End

```

### 1.2 蚁群算法设计

为进行 Job Shop 调度问题的蚁群算法解的构造,对 G&T 算法进行改进。将常规 G&T 算法在步骤中随机取冲突集  $G$  中的一个操作,改为根据信息素来计算(式(2)),按概率来选择冲突集中的操作。

$$P(o_i | \tau) = \frac{\min_{o_j \in G} \tau_{ij}}{\sum_{o_k \in G} \min_{o_j \in G} \tau_{kj}} \quad (2)$$

将在同一台机器上加工的操作  $O_i$  和  $O_j$  叫做一对相关操作,  $\tau_{ij}$  表示  $O_i$  在  $O_j$  前加工的信息素值,  $\tau_{ji}$  表示  $O_j$  在  $O_i$  前加工的信息素值。 $\tau'_{ij}$  表示信息素  $\tau_{ij}$  更新后的值。采用解  $S_{rb}$  方式更新信息素,计算方法为

$$\tau'_{ij} = \tau_{ij} + \rho \cdot (\delta(o_i, o_j) - \tau_{ij}) \quad (3)$$

其中,  $\delta(o_i, o_j) = \begin{cases} 1 & \text{如果在 } S_{rb} \text{ 中 } O_i \text{ 在 } O_j \text{ 前被加工} \\ 0 & \text{如果在 } S_{rb} \text{ 中 } O_i \text{ 在 } O_j \text{ 后被加工} \end{cases}$

当更新信息素后,还要检测更新后的信息素值是否越界。如果大于最大值,将其设置为最大值;如果小于最小值,将其设置为最小值。

### 1.3 基于关键工序的邻域交换方法

所谓关键路径是指一个可行调度中工序间无时间间隔的最长路径,可以将关键路径上的工序分解为块的序列。所谓块是关键路径在相同机器上,加工的相邻工序的集合。可交换邻域是指交换块中的前 2 个工序或后 2 个工序所组成的邻域,当块中的工序少于 2 个工序时,则无可交换邻域。

**定义 1** 令  $J_p(v)$  和  $J_n(v)$  表示工序  $v$  在工件中的前一道工序和后一道工序,  $M_p(v)$  和  $M_n(v)$  表示工序  $v$  在机器上的前一道工序和后一道工序,  $St(v)$  和  $Et(v)$  表示工序  $v$  的加工开始时间和结束时间。如果关键路径中满足  $St(v) = Et(M_p(v)) = Et(J_p(v))$ , 则该工序  $v$  称为关键工序。

在图 1 中,工序(1,5)是该解的关键工序,它依赖于工序(1,4)和(3,4),即

$$St(1,5) = Et(1,4) = Et(3,4)$$

由图 1 可以看出,当翻转到关键工序(1,5)之前的工序(2,4)与(1,4)时,由于工序(1,5)还依赖于工序(3,4),因此并不能缩短关键路径。这样就可以不考虑关键工序之前

的交换操作,提高了算法的效率。

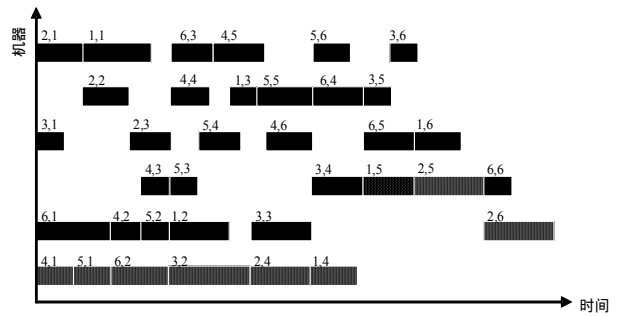


图 1 FT06 问题可行解的 Gantt 图

基于关键工序的邻域选择方法为:当关键路径中不存在关键工序时,可选择关键路径上的可交换邻域;当存在关键工序时,首先在最后一道关键工序和最后一道工序之间选择可交换邻域(可减少无效的搜索);如果此区间无可交换邻域,则结束选择(同时终止本次 TS 搜索)。采用上述邻域选择方法的 TS 算法进行局部优化,TS 算法的搜索过程经过一定步数之后,仍不能得到改进的解,则 TS 搜索结束。

## 2 实验与分析

本文实验的运行环境是:操作系统为 Windows XP, 512MB 主存, CPU 2.4GHz。程序设计语言为 C 语言。实验中采用了 JSP 典型问题(benchmarks)中较难的 13 个问题<sup>[6]</sup>,包括 FT10、LA02、LA19 问题等,并对每个问题实例分别采用 PGA(并行遗传算法)、TSAB 算法<sup>[7]</sup>和本文 TSANT 算法进行计算和比较。

试验参数如下:TSANT 算法蚁群大小为  $|O| \times 30\%$ 。其中,  $|O|$  为工序总数,进化代数数为  $|O|$ ,进行 TS 搜索的概率为 20%,TS 的搜索经过 50 步后,仍不能得到更好的解则结束该过程。其中  $\rho$  是  $[0,1]$  之间的一个随机数,通过试验结果分析,选择  $\rho = 0.1$ 、 $\tau_{\min} = 10$ 、 $\tau_{\max} = 1000$ 。信息素的初始值均设为 500;PGA 种群大小均为 200,均进化 200 代,交叉概率为 0.85,变异概率 0.05;TS 采用文献[8]的 TSAB 算法,若经过搜索过程经过 3000 步后,仍不能得到更好的解,则结束 TS。

表 1 给出了 10 次实验结果的最优值和平均值。可以看出,采用本文的 TSANT 算法得到的解只有 LA36 低于 PGA 和 TSAB 算法,但所有问题累计,得到的 makespan 的平均值较并行遗传算法(PGA)提高 2.18%、较 TSAB 算法提高 2.78%。结果表明将 TS 嵌入蚁群算法,为 TSANT 算法提供有效的搜索信息,同时利用 TS 扩大了搜索范围,这种混合蚁群算法是有效的。表 1 最右列给出了 TSANT 算法 10 次求解的平均运行时间,黑体字为最优解。

表 1 10 次实验结果的最优值和平均值

问题实例	最优解 Makespa	TSANT		PGA		TSAB		时间 (s)
		最优值	平均值	最优值	平均值	最优值	平均值	
FT10(10×10)	<b>930</b>	<b>930</b>	937.6	939	960.4	941	964.3	62.58
LA02(10×5)	<b>655</b>	<b>655</b>	659.2	<b>655</b>	670.6	<b>655</b>	673.8	16.37
LA19(10×10)	<b>842</b>	<b>842</b>	863.8	<b>842</b>	872.6	<b>842</b>	879.7	61.89
LA21(15×10)	<b>1046</b>	1055	1090.5	1060	1106.7	1064	1132.1	208.36
LA24(15×10)	<b>935</b>	944	950.9	950	966.4	951	972.8	215.91
LA25(15×10)	<b>977</b>	<b>977</b>	988.5	980	993.4	988	1008.2	206.82
LA27(20×10)	<b>1235</b>	1269	1352.2	1337	1452.6	1316	1396.5	309.57
LA29(20×10)	<b>1153</b>	1235	1312.6	1258	1336.1	1235	1363.7	301.46
LA36(15×15)	<b>1268</b>	1330	1378.8	1292	1346.3	1311	1389.7	381.26
LA37(15×15)	<b>1397</b>	1415	1463.7	1444	1493.2	1476	1503.9	366.37
LA38(15×15)	<b>1196</b>	1208	1263.8	1239	1295.6	1215	1282.7	372.15
LA39(15×15)	<b>1233</b>	<b>1233</b>	1289.5	1275	1305.7	1287	1317.2	376.19
LA40(15×15)	<b>1222</b>	1229	1243.8	1271	1318.5	1286	1322.3	388.96

(下转第 222 页)