**Flexible Job Shop 项目报告**

刘丁铱 张晙优 印煜天

**1. 原理简介**

为实现Flexible Job Shop，我们组的算法采用了Master-Apprentice Evolutionary (MAE), Path Relinking (PR)与Tabu Search (TS)方式。下面将简单分析一下各算法的原理以及它们在我们算法中的作用。

（一）Master-Apprentice Evolutionary （MAE）

算法的核心是MAE。MAE是模仿学徒进化为大师的过程的算法——两个学徒随着几代循环进化之后可以当大师，保留着前一次循环的历史。

텍스트이(가) 표시된 사진

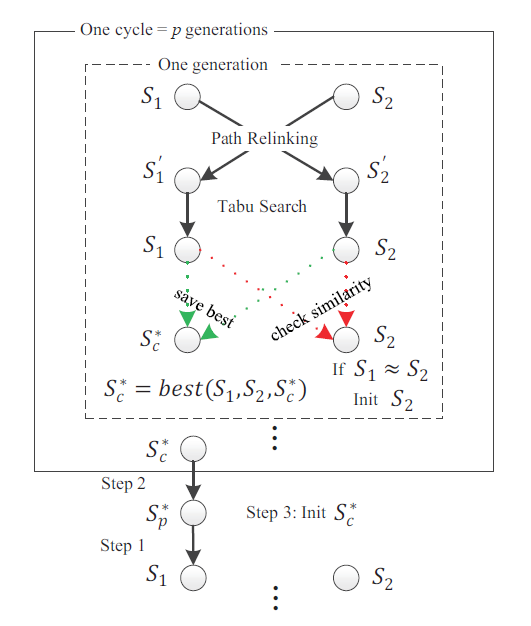
자동 생성된 설명

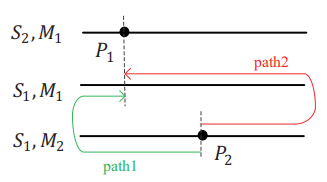
图 2 MAE算法伪代码

图 1 MAE算法流程图

通过图1和图2的伪代码可以看出MAE算法大致的流程。算法主要有三个部分组成：用于生成随机解的Init（）函数、用于改进解的禁忌搜索过程的TS（S）以及用于生成两个子解的基于路径重链接的重组算子。每一代被划分为长度为p的循环，其中p是整数。当前（上一个）循环中的最佳解决方案存储在S\*C和S\*P。开始时，MAE生成两个随机解S1和S2。然后，在每一代中，它在S1和S2通过路径重链接的重组算子来生成两个子解 S1’和 S2’，然后通过禁忌搜索过程对其进行优化，生成新的S1和S2。如果新的S1和S2比更优，则更新 S\*C 。 在每个循环结束时，在开始下一个循环之前，S1被前一个循环中找到的最佳解S\*P替换，S\*P被 S\*C替换，S\*C被设置为随机解。一旦S1接近S2（为代码中的“约等于”部分，我们代码实现为两个解有80%以上工序安排相同）,就用随机解替换S2，以确保搜索的多样性。最后，返回搜索过程中找到的最佳解决方案S\*。

在我们的代码中，我们发现gen代数对解的质量有影响，分别测试了gen=15，10和5，发现gen=5左右的时候每个cycle得到的解的质量已经趋于稳定，所以取了gen=5的值以增加cycle次数。

1. Path Relinking (PR)

我们算法中采用了Path Relinking技术来找出更优的解。在上面的MAE为代码中，对两个解S1和S2，PR(S1,S2)通过交换工序顺序，找到比S1更接近 S2的解。重连接过程中，解之间的“距离”的定义很重要。我们如下定义某一道工序的距离：

텍스트, 시계이(가) 표시된 사진

자동 생성된 설명

图 4

图 3

如果某工序在S1,S2中在同一台机器上执行，距离定义为位置之差的绝对值，如图3的。若在不同机器上执行，距离定义为Path 1与Path 2中较小的一个，如图4。最后， S1与S2之间的距离定义为所有工序距离之和。

我们采用的路径重连接算法的伪代码如图5。可见主要参数有α，β，γ。我们的参考论文中分别取了0.4，0.5与5，我们代码从这些值开始调整发现调整之后效果不是很明显，决定直接采用这些值。

*텍스트이(가) 표시된 사진

자동 생성된 설명*

图 5 路径重连接过程伪代码

（三）Tabu Search （TS）

禁忌搜索是类似模拟退火算法，遗传算法的找优解算法。模拟退火算法和遗传算法一般只会基于相邻搜索找最优，经常发生卡在局部最优解的情况。为避免这种情况，禁忌搜索会存储局部最优解的信息并躲开它来找最优解。

在MAE中调用禁忌搜索过程TS（S），以加强搜索。它通过将关键操作重新分配给不同的机器并将其插入到可行的位置，或通过更改同一机器上关键操作的位置来改进解决方案。我们代码从邻域中产生候选解，这里邻域的选取是重点。我们定义关键路径中若干连续工序在同一机器上加工，则称这些工序为关键块，关键路径时加工开始到加工结束所经历的路径，路径上的点代表某道工序在某机器上执行。

TS算法中的几个参数中我们调试了TS中的禁忌表长度，它过短会循环搜索，过长会陷入局部最优，大致测试后选择为5。TS迭代次数也是关键，因为我们将迭代次数作为TS终止的标志。它过小会影响解的质量，过大会浪费时间，于是测试了10000、5000、1000、500后选择为1000。

（四）插单操作

在原始订单处理方案ans及机器分配情况machine的基础上，采取贪心策略：

首先在读入插单时，对于每一道工序的可选机器根据消耗时间从小到大排序，选取消耗时间最小的机器；根据插单中上一道工序结束时间（初始化为插单时间点itime）在machine中找到该机器最早可以插单的时间点，并更新ans与machine。

这样处理主要的考虑有以下两点：一是我们所用算法的搜索结构比较复杂，难以将itime这一参数添加到算法中，因此难以使用更优的算法；二是贪心的处理时间较短，可以有更加充足的时间处理订单数更多的原始订单。正如后文中将要提到的对结果的分析，甘特图中有大量空白，因此直接使用贪心仍然能获得较好的结果。

**2. 实验结果分析**

各订单的makespan见上图。从中可见由于贪心完整保留了初始解，插单对于空余机器的利用还不够充分（随机的插单时间可能也是其原因），尤其是第一次插单后总makespan有较大幅度增长，这是此算法的不足之处。

比较好的一点是由于求初始解的时间充足，初始解的质量较高，这在一定程度上弥补了插单算法的不足。

**参考文献**

【1】 A tabu search/path relinking algorithm to solve the job shop scheduling problem,Computers & Operations Research,2015,53:154–164

【2】 A Two-Individual Based Evolutionary Algorithm for the Flexible Job Shop Scheduling Problem,AAAI19

【3】 Scatter search with path relinking for the flexible job shop scheduling problem,European Journal of Operational Research,2015,245,35-45