基于带变邻域搜索的自适应模拟退火算法解决零空闲流水线调度优化问题

汇报人：王云鹤
指导教师：殷明浩

目 录

1 研究背景与研究工作

2 零空闲流水线调度问题

3 带变邻域搜索的自适应模拟退火算法

4 实验结果

1 研究的背景与研究工作

研究背景

生产调度在实现先进制造业和提高生产效率中扮演一个基本和关键的角色。根据不同的约束条件，车间工作调度问题包括零空闲流水线调度、无等待流水线调度问题等。在现实生产调度中，有的时候一旦流水线生产开始，机器的运转是不允许停止的，这就是零空闲流水线调度问题。现在的构造式启发算法如NEH，元启发式算法如DE解决该问题时，没有很大程度地提高解的质量。

1 研究的背景与研究工作

研究工作

为了获得零空闲流水线调度问题更好的解，本文提出一种带变邻域搜索的自适应模拟退火算法解决它。首先该算法使用模拟退火算法的基本思想，其次它的参数是根据具体问题例子大小及算法进行的程度来调节的，这就是所谓的“自适用”，再次该算法提出两阶段局部搜索的策略，提高解的收敛速度以获得一个全局最优解。局部搜索算法采用的是高效率的变邻域局部搜索方法。最后用C语言实现新算法，使具体问题的最大完成时间最小化。

2 零空闲流水线调度问题

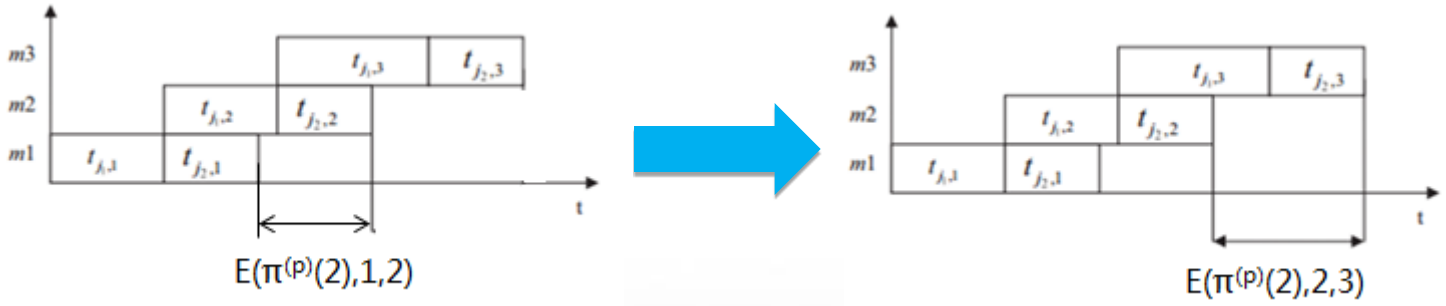
问题描述

在一个生产系统中，有n个工作和m个机器，所有工作在其它任何一个机器上的处理顺序和在第一台机器上的处理顺序是相同的，每个工作的准备时间为0或者包含在处理时间中。
要求:
1.每一个工作在某一时刻只能在一个机器上处理；
2.一台机器某一刻只能处理一个工作；
3.机器上的工作一旦开始，那么机器的运转是不可以停止的。
所谓的零空闲，即每个机器上的操作处理工作阶段是连续的，不可以停止。

零空闲流水线调度问题

具体实例

当n=2,m=3时，对于工作调度序列为：$\pi=\{\pi_1,\pi_2\}$,
如何计算出最大完成时间makespan的大小。



makespan=$t_{j1,1}+t_{j2,1}+E(\pi^{(p)}(2),1,2)+E(\pi^{(p)}(2),2,3)$

3 带变邻域搜索的自适应模拟退火算法

参数设置

参数	意义	值
N	工作个数	样例中大小
M	机器个数	样例中大小
Initial_t	初始温度	1000.0
cool_t1	第一阶段温度冷却系数	0.99999
cool_t2	第二阶段温度冷却系数	0.9999
vns_t1	第一阶段变邻域搜索循环次数	N/3
vns_t2	第二阶段变邻域搜索循环次数	3×N/2
end_t1	结束温度	0.0025
end_t2	第二阶段变邻域搜索的开始温度	0.005
current_t	当前温度	current_t=current_t×cool_t

3 带变邻域搜索的自适应模拟退火算法

vns算法

算法流程：

1. 输入一个工作调度序列作为初始序列；
2. 交换初始工作调度序列的两个位置，产生一个新的工作调度序列，即为当前序列；
3. 当循环次数没有达到终止条件时执行4-8；
4. 计数变量初始化为0；
5. 当计数变量小于2时，执行6-8；
6. 当计数变量为0时，对当前序列执行插入操作，形成新序列；
7. 当计数变量为1时，对当前序列执行段插入操作，形成新序列；
8. 如果新序列比当前序列适应度高，即表现为最大完成时间比较少，则计数变量重新赋值为0，更新新序列为当前序列；否则计数变量加1；
9. 选择当前序列和初始序列中较优的序列，作为最优序列；
10. 输出最优序列和它的适应度。

3 带变邻域搜索的自适应模拟退火算法

ASAvns算法

算法流程：

1. 随机产生一个工作调度序列作为初始序列，假设最初时初始序列为最优序列，计数变量初始化为0；
2. 如果当前温度高于结束温度或者没有到达结束条件，执行3-10；
3. 如果该语句第一次执行而且当前温度小于第二阶段开始温度，初始化最初温度，第二阶段冷却系数，第二阶段变邻域搜索次数；
4. 初始化最优序列为当前序列；
5. 当循环次数没有到达第一或者第二阶段变邻域搜索次数时，执行6-8；
6. 当执行时间大于时间段上限时，跳出循环；
7. 对当前序列执行vns算法，获得一个新序列；
8. 如果新序列适应度高于当前序列，更新新序列为当前序列；
9. 如果当前序列适应度高于最优序列，更新新序列为最优序列，否则计数变量加1；如果计数变量为N的倍数时，强制接受新序列为最优序列；
10. 更新当前温度；
11.输出最优序列及它的适应度。

3 带变邻域搜索的自适应模拟退火算法

ASAvns算法创新性

1.自适应

自适应参数调整，参数随着零空闲流水线调度具体问题例子大小及算法进行的程度来调节，如搜索循环次数和强制接受新解的条件。

2.变邻域搜索

该算法将两种简单有效的搜索方法包括插入和段插入局部搜索方式，嵌入到变邻域搜索算法基本框架里，形成一种高效、独特的变邻域搜索算法。

3.两阶段搜索

算法开始时，执行第一阶段的搜索，使用变邻域搜索算法获得一些可行解。当温度降低至0.005时，第二阶段搜索开始进行。循环变邻域搜索过程，收获在第一阶段获得解基础上更好质量的解，高效率地获得更优的工作调度序列。

4 实验结果

对比1

Problem	Scale	NEH	ASAvns	PIP
	n×m	Makespan	Makespan	
Ta001	20×5	1 413	**1 380**	**-2.335%**
Ta011	20×10	2 266	**2 188**	**-3.442%**
Ta021	20×20	3 482	**3 205**	**-7.955%**
Ta031	50×5	3 028	**3 014**	**-0.462%**
Ta041	50×10	3 579	**3 426**	**-4.275%**
Ta051	50×20	5 813	**5 408**	**-6.967%**
Ta061	100×5	5 874	**5 836**	**-0.647%**
Ta071	100×10	6 893	**6 756**	**-1.988%**
Ta081	100×20	9 667	**9 367**	**-3.103%**
Ta091	200×10	11 926	**11 684**	**-2.029%**
Ta101	200×20	15 330	**15 004**	**-2.127%**
Ta111	500×20	30 600	**30 049**	**-1.801%**
Average	PIP:	**-3.094%**		

4 实验结果

对比2

Scale	IG	KK	ASAvns		
n×m	ARE	ARE	ARE	ASD	AT
20×5	9.19	0.87	**-3.15**	0.23	1.0S
20×10	8.37	2.00	**-7.73**	0.08	1.0S
20×20	5.40	1.29	**-7.28**	0.03	1.0S
50×5	11.53	-0.03	**-1.37**	0.02	5.0S
50×10	12.34	-1.79	**-6.12**	0.03	5.0S
50×20	12.44	-0.55	**-7.58**	0.03	5.0S
100×5	16.40	-0.24	**-1.21**	0.01	10.0S
100×10	13.98	0.08	**-2.45**	0.01	10.0S
100×20	14.78	-2.57	**-5.47**	0.01	10.0S
200×10	17.18	-0.55	**-2.15**	0.00	20.0S
200×20	16.38	-2.23	**-4.15**	0.01	21.0S
500×20	18.68	-1.65	**-1.18**	0.00	67.4S
Average	13.055	-0.447	**-4.15**	0.04	13.07S

统计参数：平均相对差错百分比ARE（the average relative percent error），平均标准差（the average standard deviation）和平均时间AT（the average time）。当ARE越小时，表示算法对于具体问题结果就越好。

4 实验结果

对比3

Scale	PSOvns			HDPSO			ASAvns		
n×m	ARE	ASD	AT	ARE	ASD	AT	ARE	ASD	AT
20×5	-3.01	0.04	2.0S	-3.03	0.00	2.0S	**-3.15**	0.23	1.0S
20×10	-7.16	0.15	2.0S	-7.28	0.02	2.0S	**-7.73**	0.08	1.0S
20×20	-6.81	0.27	2.0S	-7.04	0.03	2.0S	**-7.28**	0.03	1.0S
50×5	-0.75	0.00	5.0S	-0.75	0.00	5.0S	**-1.37**	0.02	5.0S
50×10	-5.75	0.22	5.0S	-5.88	0.10	5.0S	**-6.12**	0.03	5.0S
50×20	-6.65	0.42	5.0S	-7.27	0.21	5.0S	**-7.58**	0.03	5.0S
100×5	-0.72	0.03	10.0S	-0.73	0.00	10.0S	**-1.21**	0.01	10.0S
100×10	-1.40	0.11	10.0S	-1.55	0.03	10.0S	**-2.45**	0.01	10.0S
100×20	-4.66	0.33	10.0S	-5.36	0.08	10.0S	**-5.47**	0.01	10.0S
200×10	-1.41	0.12	20.0S	-1.54	0.02	20.0S	**-2.15**	0.00	20.0S
200×20	-3.23	0.31	20.0S	-3.71	0.13	20.0S	**-4.15**	0.01	21.0S
500×20	-1.85	0.15	50.0S	-2.16	0.10	50.0S	-1.18	0.00	67.4S
Average	-3.62	0.18	11.75S	-3.86	0.06	11.75S	**-4.15**	0.04	13.07S

由于ARE的平均值高于HDPSO和PSOvns算法, 可以说ASAvns算法显著提高了算法性能，优化了零空闲流水线调度具体问题计算结果。该算法比PSOvns算法和HDPSO算法有着更优越的效率和更强的鲁棒性。

谢 谢 大 家!