**成绩**



**最优化方法**

**作业报告**

**题　　目： 港口调度问题的贪心算法和**

**元启发算法（模拟退火SA）**

**实现**

**学 院： 计算机学院**

**专业名称： 计算机科学与技术**

**学 号： 1120172988**

**姓 名： 刘若轩**

**任课教师： 李冬妮**

**评 阅 人：**

港口调度问题的贪心算法和元启发算法（模拟退火SA）实现

刘若轩 (学号: 1120172988)

摘 要: 为了解决港口调度问题，高效地计算出港口调度问题的最优策略，比较了两种不同的贪心策略和一种模拟退火算法（SA）的解，在有限时间内得出该港口调度问题的近似最优解，充分发挥了模拟退火算法搜索面广的优点。实验中采用了两种贪心策略，模拟退火算法（SA）相较于贪心算法引入了随机因素，避免了贪心算法陷入局部最优解的缺点，以一定概率接受较差解，在时间趋向于正无穷的搜索时间内，模拟退火算法计算出最优解的概率趋近于1。结果表明模拟退火算法（SA）在该港口调度问题中运算速度快，最优解概率大，优于两种贪心策略，可为港口调度问题提供有效支持。

关键词: 贪心算法;模拟退火算法;港口调度问题

# 引言

a) 问题背景

港口调度问题是一道具体、综合而实际的调度问题，而本题中的港口调度是一道混合流水车间调度问题，是典型的NP-hard问题。先前对于港口调度问题的相关研究包括对于港口调度中泊位分配和港口资源分配的研究：ETSUKO[1]等对港口中泊位的分配问题进行研究，探讨泊位分配中的优先级问题，王中华[2]和王冠儒[3]在泊位调度模型的基础上改进遗传算法（GA）求解在港的轮船调度问题，实现货轮的自动分配。

在本题中，针对具体的港口调度问题，分析贪心策略的缺点，寻找合适的元启发算法，并计算近似最优值。

b) 问题描述

港口调度问题要解决的是货轮开始入泊到所有货轮出泊完成，且所有货物起吊到小车运送的优先顺序问题。

在本题中，港口调度问题分为三个子问题，泊位分配问题，起吊机分配问题和小车分配问题。

泊位分配问题：港口有两个泊位，四艘货轮同时入港，如何安排入泊顺序，使得所有货轮出泊完成，且小车运送完所有货物用时最短。

起吊机问题：港口有两台起吊机，在泊位分配问题的基础上，如何安排货物起吊顺序，使得所有货轮出泊完成，且小车运送完所有货物用时最短。

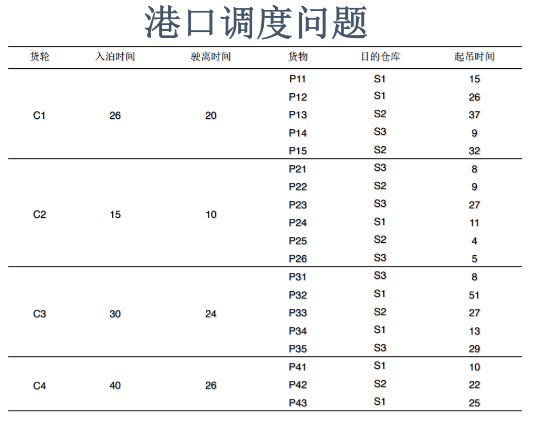
 小车分配问题：港口有两辆小车，在起吊机问题的基础上，如何安排货物运送顺序，所有货轮出泊完成，且小车运送完所有货物用时最短。

图1 港口调度问题条件

三个问题单独考虑时满足贪心性质，且最优子结构性质，以并行车间调度的模型皆可解决三个问题。然而混合车间调度存在工件加工顺序的约束条件（在本题中即为货物随货轮入泊、起吊、运送的固定顺序），因此需要考虑偏后工序的机器处于空闲状态，存在资源的浪费。因此港口调度问题的三个子问题不可单独分开讨论，需要在图1条件下综合考虑。

c) 方案及实验效果

**贪心策略一**

泊位分配：先小后大

起吊机分配：先大后小，分船起吊

小车分配：先来先运

在第一种贪心策略中，考虑到由于货轮入泊、出泊时间较长，对资源空闲带来的影响较大，采取入泊和出泊时间较短的货轮优先级高的策略，而货物中起吊时间差异较大，可能造成并行车间调度模型中的大时间差现象，且分船运输减弱了两艘货轮同时入泊出泊造成的起吊机空闲的后果的影响，因此采用先大后小，分船起吊的策略。在本实例中，该贪心算法求出的解为230。

**贪心策略二**

泊位分配：先小后大

起吊机分配：先小后大

小车分配：先大后小

第二种贪心策略中，泊位分配和起吊机分配采取同样的分配策略，时间段的优先级高，使小车能在最短的时间内投入工作，而小车分配采取先大后小的运输策略，即并行车间分配的策略。在本实例中，该贪心算法求出的解为255.

**模拟退火算法（SA）**

在模拟退火算法中引入了随机因素，把较优的贪心策略一得到的解作为初始解，首先产生函数随即交换当前解的优先级，从当前解（可能）随机产生一个解空间的新解（因为工序靠后的调度不仅依赖于货物的优先级，还取决于先前的调度结果，因此，单纯的交换优先级可能不会影响调度顺序。解决方案是调整参数，以较多迭代次数计算求最优解，忽略计算资源的浪费）；第二步计算与新解所对应的[目标函数](https://baike.baidu.com/item/%E7%9B%AE%E6%A0%87%E5%87%BD%E6%95%B0)差；第三步判断新解是否可以被接受，第四步是若新解被确定接受时，用新解代替当前解。经过对温度参数初值，迭代条件和衰减因子的调整，得以在一定时间内以较大概率求得近似最优解。在本实例中，该模拟退火算法随机进行100次计算，100次解得266，0次解得其他，则266较大概率为次港口调度问题的最优解。

# 算法设计

2.1贪心策略：

2.1.1目标函数

构建三个队列分别用于保存待入泊货轮freeship、待起吊货物freegoods和待运输货物buffergoods，构建三个向量分别用于保存在泊的货轮parkingship以及在吊craninggoods和在运shuttlinggoods的货物。按优先级w条件关系，定义比较函数，用于队列内部排序。当状态state要发生转换时，按照自定义优先级关系，从队列首或队列尾取出对象进行状态转换。时间变量t循环进行累加，依此判断是否在泊（入泊完成、装卸完成、出泊完成）、入泊（开始入泊）、在吊（起吊完成）（若一船装卸完成，返回判断装卸完成并继续循环）、起吊（开始起吊）、在运（运输完成）、起运（开始运输）。当所有的货物均到达各自目的仓库且所有货轮均驶 离泊位且所有小车均返回缓冲区时停止循环，得到当前解的值，其过程如下：

1.令初始freeship为输入条件，时间t=0

2. i为下一状态，lifeline为时间节点，若lifeline[i]=t，依此进行状态转换

3.若未满足停止循环条件，t=t+1，返回2

4.否则，停止，t为当前解

2.2模拟退火算法（SA）：

2.2.1模拟退火

设定模拟退火阈值L，在温度tk下，对当前解 Vi进行 SA，其过程如下:

1．令初始当前状态S = Vi，初始最优解S' = Vi．

3．由状态S产生新状态S'，计算目标函数增量． Δc' = c(S')- c(S) ．

4．若Δc'＜0,则接受S'为当前状态,若Δc'＞0,以概率exp(-△c'/tk )接受S'为当前状态，若S'被接受，则令S' = S．

5．tk进行退温．

6．重复执行步骤 1-5，直到tk<模拟退火阈值L．

2.2.2构造新解

用随机化函数随机交换入泊优先级或起吊优先级或运输优先级，而改变优先级有可能没有产生新的解，这种情况下，仍然接受新解，扩大搜索面积。其过程如下：

1.随机数产生函数产生随机数x

2.x%3判断交换哪一类优先级

3.产生两个随机数a,b%对象总数

4.若a=b，返回3

5.交换a和b优先级

2.2.3退温过程及收敛判断

退温公式如下: tk = αk\* t0，式中: α 为 0.999，k 为降温次数，t0 为初温。

# 实验

## 实验设置

实验环境：Microsoft Visual Studio 2019

处理器 Intel® Core™ i5-7300HQ CPU@2.50GHz

RAM 8.00GB

## 实验结果

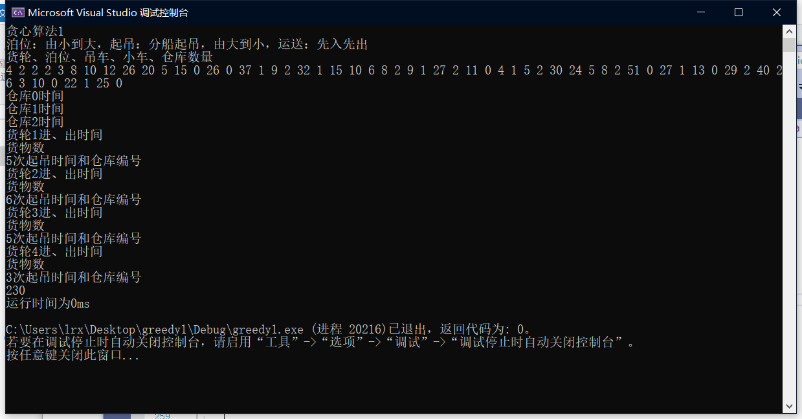
贪心策略一 十次运行结果230，平均运行时间1ms

图2 贪心策略一运行结果

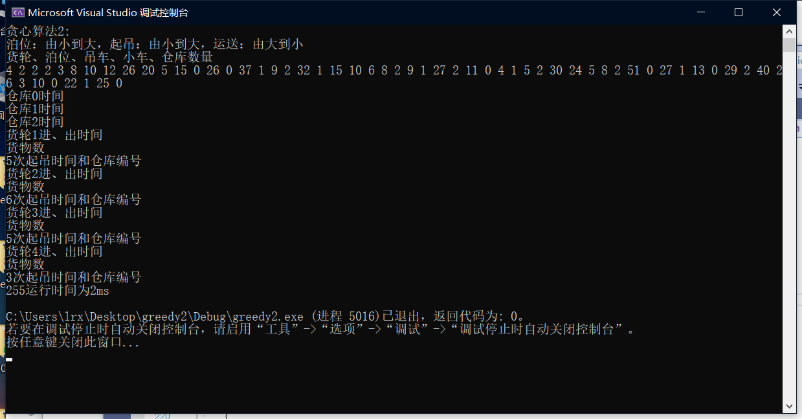
贪心策略二 十次运行结果255，平均运行时间1ms

图3 贪心策略二运行结果

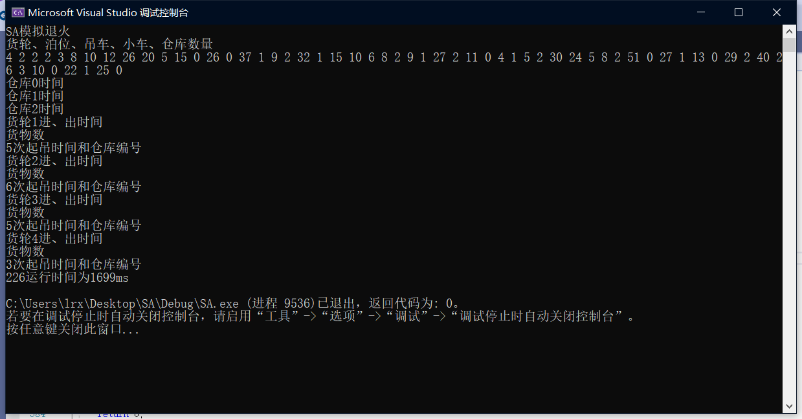
模拟退火运行结果 100次运行，求得100次时间为226，平均时间1759.9ms

图4 模拟退火运行结果

图5 模拟退火 货物时序图

表**1** 运行结果与运行时间

|  |  |  |
| --- | --- | --- |
| 算法 | 运行结果 | 运行时间（秒） |
| 贪心1 | 230 | 0.001 |
| 贪心2 | 255 | 0.001 |
| SA(1e5,1e-16,0.999) | 226 | 1.7599 |

# 总结

从以上结果来看，模拟退火算法求得226的频率是100/100，较大概率确认226为该实例最优解。而从调度过程来看，好的贪心策略可以达到接近最优解，因此，当元启发计算时间较长，而结果精度无需非常高的情况下，我们可以采用贪心解。经过对SA仿真，我发现SA在 110代左右收敛，收敛性较好，但由于SA是随机化的贪心思想，不一定每次都能求得最优解。通过调整衰减因子或退火温度阈值，调整迭代次数，以适应不同实例下的精度要求，或是改进SA，记忆改进解方向，避免向更低处搜索，浪费计算资源。

References:

1. ETSUKO N，IMAIA，PAPADIMITRIOU S．Berth Allocation Planning in the Public Berth System by Genetic Algorithms[J]．European Journal of Operational Research，2001(131)：282-292．
2. 王中华．基于遗传算法的港口船舶调度优化问题研究[D]．上海：上海海事大学，2007．
3. 王冠儒．基于遗传算法的VTS船舶调度的设计与实现[D]．大连：大连海事大学，2012．