用模拟退火算法解决单机调度问题

王圣富

- 北京理工大学计算机学院 2016 级
- 学号：1120161848
- 班号：07111605

摘　要

为了解决单机调度中工件排序的问题，确保目标函数调度总延迟最小，根据单机调度问题的特点，进一步将其转化为排序优化问题并利用模拟退火算法进行求解。模拟退火算法中使用了随机选取已有解中的两个位置，交换其内容得到新解的策略，降温采用几何方式。对一个具有 8 个工件的单机调度问题进行了优化计算，得到了与穷举相同的最优解，验证了模拟退火是求解单机调度问题一种有效算法。

关键词

单机调度；
局部搜索；
模拟退火；
蒙特卡洛；
甘特图

§1 引言

在生产调度研究领域，单机调度问题（Single Machine Scheduling Problem），是生产运营管理领域研究的重要课题。进行作业调度问题的研究对于有效缩短产品加工时间，降低产品生产成本，提高企业生产效率等方面有重要的实际应用价值。

局部搜索，是解决最优化问题的一种元启发式算法。局部搜索从一个初始解出发，然后搜索解的邻域，如有更优的解则移动至该解并继续执行搜索，否则返回当前解。虽然基本的局部搜索算法易于实现，但容易陷入局部最优且解的质量与初始解和邻域的结构密切相关。所以本文使用局部搜索的改进方法——模拟退火。

模拟退火算法（Simulate Anneal，SA）是一种通用概率演算法，用来在一个大的搜寻空

间内找寻命题的最优解。其原理和金属退火的原理近似：将热力学的理论套用到统计学上，将搜寻空间内每一点看做空气内的分子；分子的能量，就是它本身的动能；而搜寻空间内的每一点，也像空气分子一样带有“能量”，以表示该点对命题的合适程度。演算法先以搜寻空间内一个任意点作起始：每一步先选择一个“邻居”，然后再计算从现有位置到达“邻居”的概率。

本文主要研究一种具有序列相关的带准备时间的单机调度问题，调度目标就是要找到一个最优的工作序列使得总延迟最小。对于排序问题，我们有回溯、分支限界等等算法，但是解空间极大。所以我们使用基于蒙特卡罗方法的模拟退火算法，因为我们需要对单机可能的调度进行分析性计算。对于那些由于计算过于复杂而难以得到解析解或者根本没有解析解的问题，蒙特卡罗方法是一种有效的求出数值解的方法。本文同时使用甘特图进行可视化，使调度更加直观、易于理解。

§2 模型假设

2.1 背景

1. 每个工件上只可被机器加工一次，且只有在工件到达机器处后才可以被加工。
2. 每个机器某一个时刻最多只能执行一个工件，而且执行过程是非抢占的。

2.2 符号

R_i：　工件i的释放时间；

S_i：　工件i的实际开工时间；

P_i：　工件i的加工时间；

D_i：　工件i的计划完工时间；

d_i：　工件i的实际完工时间与计划完工时间之差；

U_i：　如果工件i在某调度计划中延误，$U_i = 1$，否则$U_i = 0$；

ΔE：　评价值增量（温度差）；

$P(e)$：　接受函数；

$T(t)$：　冷却进度；

§3 算法设计

3.1 退火步骤

模拟退火算法可以分解为解空间、目标函数和初始解三部分。其基本思想大致如下：

1) 初始化：初始温度 T(充分大)，初始解状态 S(算法迭代的起点)， 每个 T 值的迭代次数 L
2) 对 k=1，……，L 做第(3)至第 6 步：
3) 产生新解 S′
4) 计算增量 Δt′=C(S′)-C(S)，其中 C(S)为评价函数
5) 若 Δt′<0 则接受 S′作为新的当前解，否则以概率 exp(-Δt′/T)接受 S′作为新的当前解.
6) 如果满足终止条件则输出当前解作为最优解，结束程序。终止条件通常取为连续若干个新解都没有被接受时终止算法。
7) T 逐渐减少，且 T->0，然后转第 2 步。

3.2 状态接受函数

具体地，接受概率P可以写成如下形式的函数：

$$P=\begin{cases}1,\Delta E<0\\ \exp(-\Delta E/T),\Delta E\geq 0\end{cases}$$

可以看出，接受函数$P(e)$与评价值增量ΔE和当前温度T有关．对于$\Delta E<0$的情况，接受概率P恒等于 1。对于$\Delta E\geq 0$的情况：在ΔE相同的情况下，T越高，接受概率P越大，T越低，接受概率P越小；在当前温度T相同的情况下，ΔE越大，接受概率P越小，ΔE越小，接受概率P越大。

使用模拟退火算法，在高温条件下，即计算的初期，可能会接受比当前最优调度序列评价值差的调度序列作为新的最优调度序列；在低温条件下，即计算的后期，基本只接受比当前最优解评价值好的新调度序列作为新的最优调度序列。这种特性令模拟退火算法不会陷入局部极值，因而可以进行全局优化。

3.3 冷却进度表

快速模拟退火算法如下：

$$T(t)=T_0/(t+1)$$

这是因为经典模拟退火算法的冷却进度表的计算量太大，因此在文中使用快速模拟退火算

法的冷却进度表。

3.4 初始温度

温度的高低会影响接受概率的大小。在确定初始温度时，常用的方法有：随机法，经验法。本文采用经验法，将初始温度设为工件数量的 100 倍。

3.5 算法终止准则

模拟退火算法中常用的终止准则包括：

1. 设置循环迭代的次数；
2. 计算过程中，连续许多步不接受新解；
3. 设置终止温度，该终止温度要足够低，以便得到稳定解。

文中采用第 3 种终止准则，即给定终止温度。

3.6 设计

首先进行初始化，设定初始温度、内部蒙特卡洛循环迭代次数和冷却系数等，利用随机置换函数创建一个随机调度序列。如下：

```
temperature=100*numOfJobs;%初始温度
ite=100;%内部蒙特卡洛循环迭代次数
temDropFactor=0.99;%冷却系数
permutation=randperm(numOfJobs);
```

设定停止迭代温度为 0.001，在每一次迭代扰动中，先计算原调度总延迟，经过随机扰动之后，再计算新调度总延迟。其中，新老调度的差值，就相当于能量。

```
curdelay=calculate(permutation,jobs);%计算原调度总延迟
newPermutation=perturb(permutation);%产生随机扰动
newdelay=calculate(newPermutation,jobs);%计算新调度总延迟
delta_e=newdelay-curdelay;%新老调度的差值，相当于能量
```

若新调度好于旧调度，就用新调度代替旧调度。新调度差于旧调度时，温度越低，越不太可能接受新解，新老调度差值越大，越不太可能接受新解。所以，用概率选择是否接受新解，

尽管可能得到较差的解。

```
if delta_e<0
    permutation=newPermutation;
else
    if exp(-delta_e/temperature)>rand()
        permutation=newPermutation;
    end
end
```

接着，迭代次数加一，进行必要的更新操作和可视化操作（如甘特图）

```
cnt=cnt+1;
delay(cnt)=calculate(permutation,jobs);
temperature=temperature*temDropFactor;
```

§4 编程实验

4.1 前期设置

模拟退火算法虽然易于理解，但需要大量矩阵运算和更新操作，C++作为面向对象操作语言不能很好地实现该算法，所以本文选择 Matlab R2018a 作为编程环境。

模拟退火算法作为局部搜索的改进随机算法，对于不同的初始温度、迭代次数和下降系数有着不同的正确率。初始温度 T 选择越高，则搜索到全局最优解的可能性也越大，但计算复杂度也显著增大。在本次实验中，我们利用 Matlab 高效的矩阵运算和内存处理使得时间控制在 10 秒以内。

硬件方面，本人使用的电脑型号为 Lenovo Yoga 710-14，Windows10 2018 春季创意者版，超极本类别，6 代中央处理器。

4.2 结果

首先我们运行编写好的程序，得到如下结果。其中，中文数字表示工件排在调度序列的第几位。

表 1：模拟退火算法的结果

样例	01	02	03	04	05
退火解	58	367	3053	5628	9711
一	1	1	2	14	2
二	2	2	3	13	5
三	3	4	4	16	12
四	4	3	7	5	20
五	5	7	13	15	21
六		9	12	11	13
七		10	14	17	11
八		8	11	1	25
九		6	1	10	22
十		5	9	2	17
十一			10	4	9
十二			15	18	14
十三			8	6	3
十四			5	20	18
十五			6	8	15
十六				7	19
十七				19	7
十八					6

根据已经得到的针对五个样例的单机调度，我们可以制作如下的甘特图。其中，横坐标代表工件的加工时间。

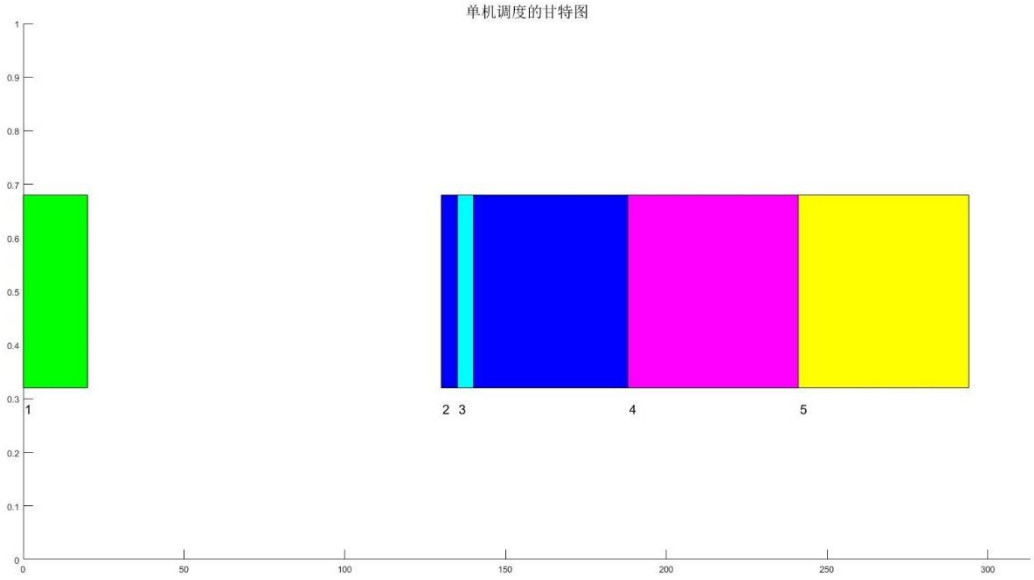

图 1：用模拟退火解样例一

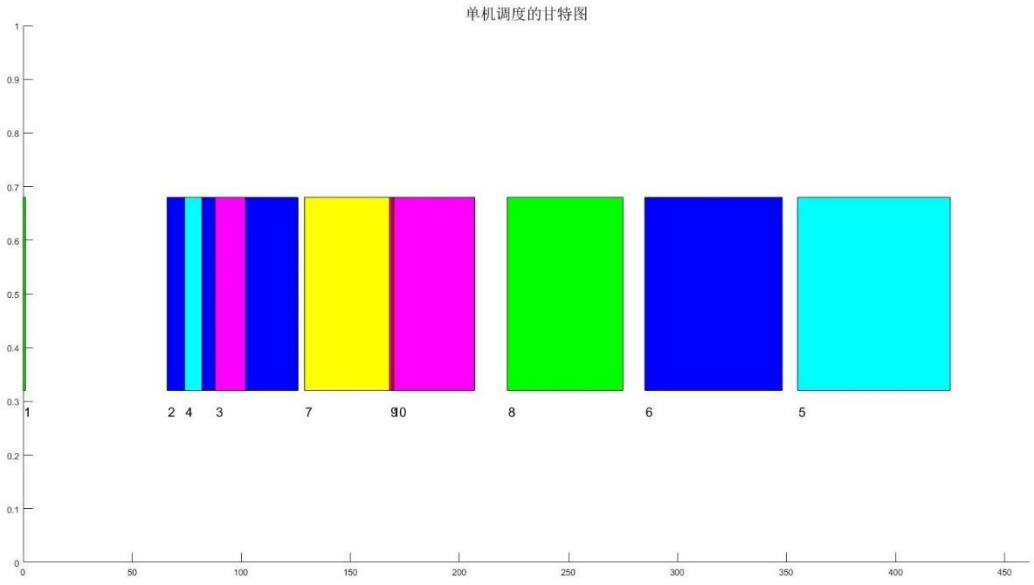

图 2：用模拟退火解样例二

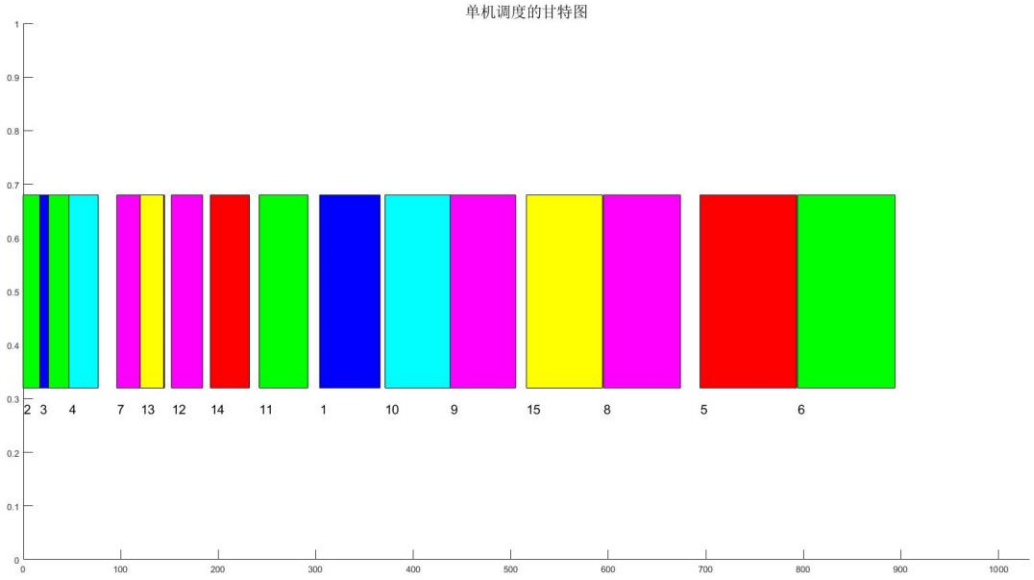

图 3：用模拟退火解样例三

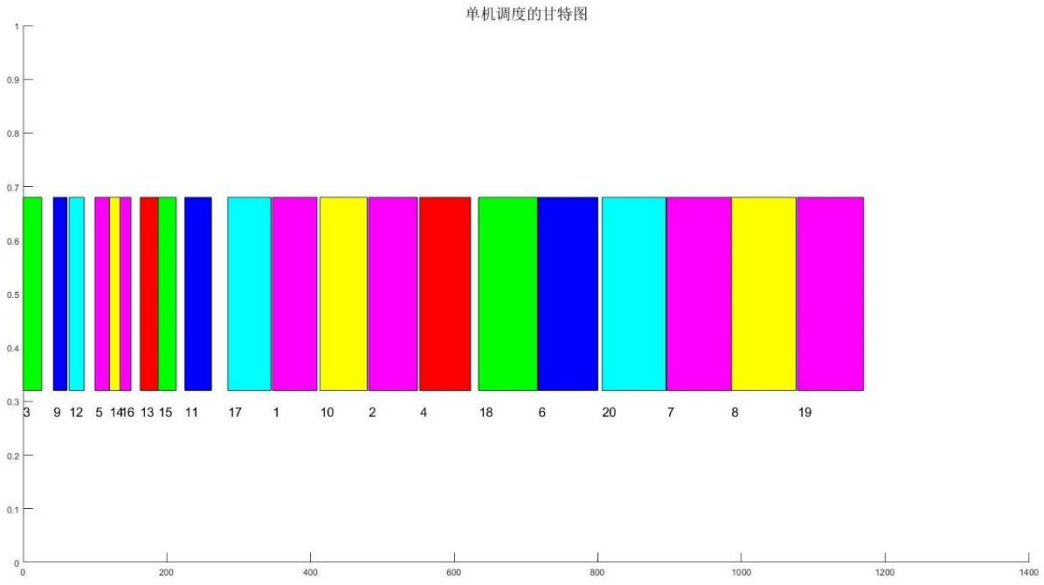

图 4：用模拟退火解样例四

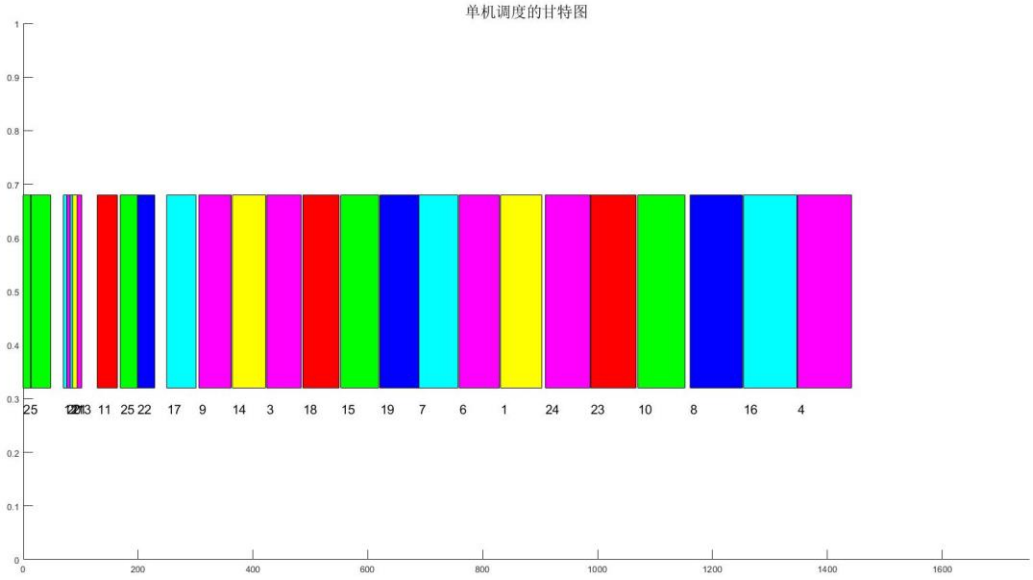

图 5：用模拟退火解样例五

为了充分了解模拟退火算法在迭代退火的训练过程当中，目标函数总延迟的变化情况，我们需要单机调度的训练过程图。每一个样例的模拟退火训练过程如下：

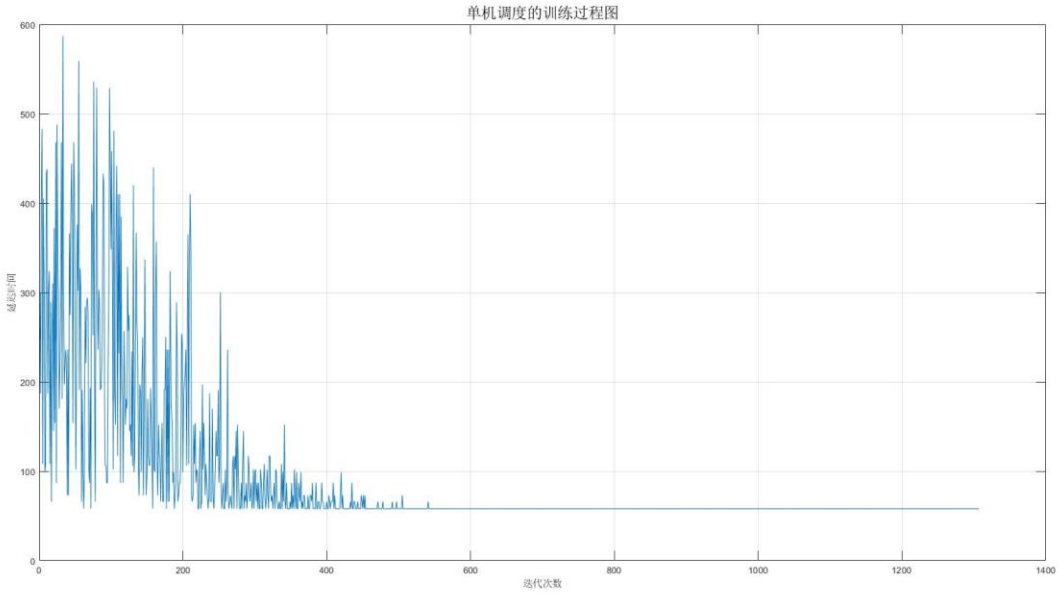

图 6：样例一的延迟时间变化

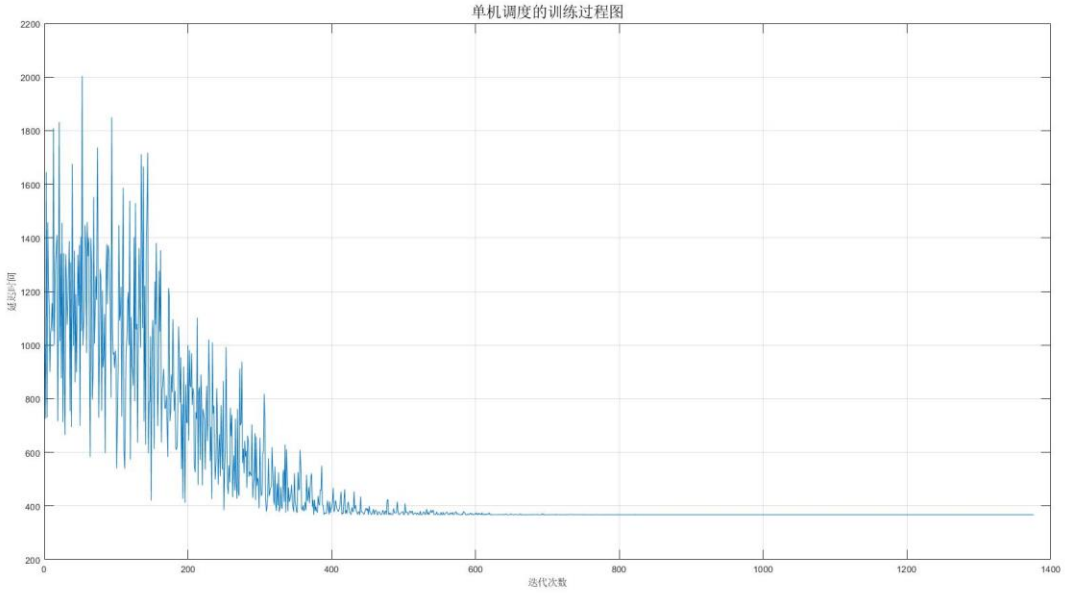

图 7：样例二的延迟时间变化

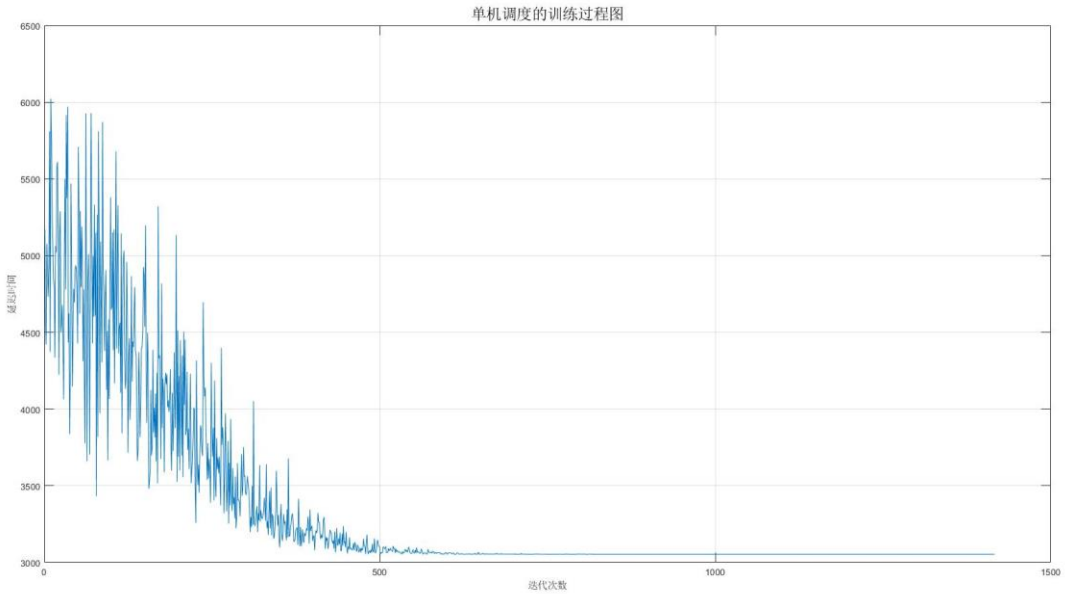

图 8：样例三的延迟时间变化

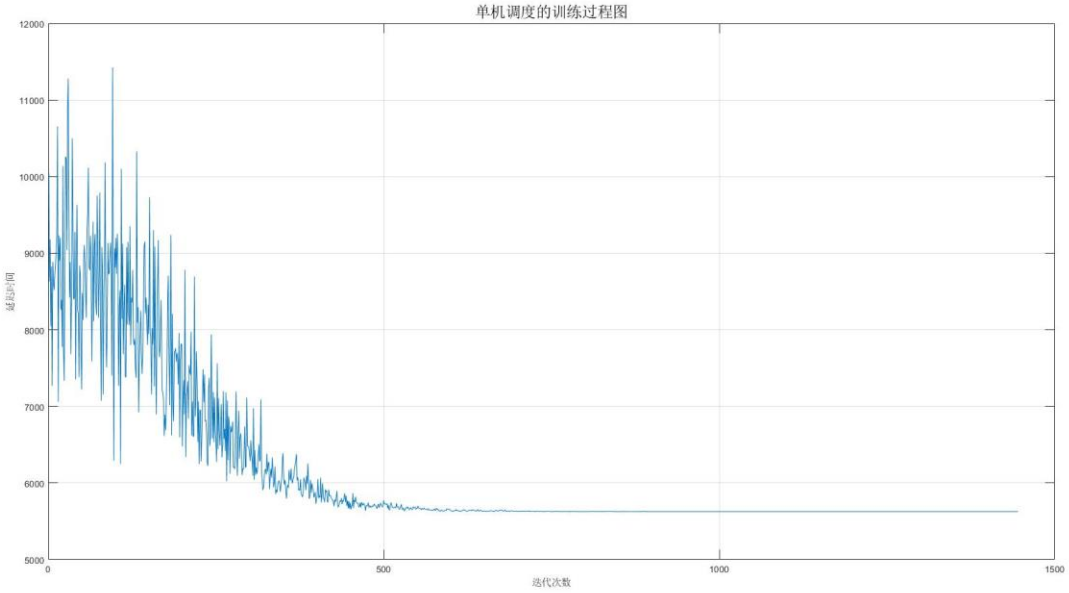

图 9：样例四的延迟时间变化

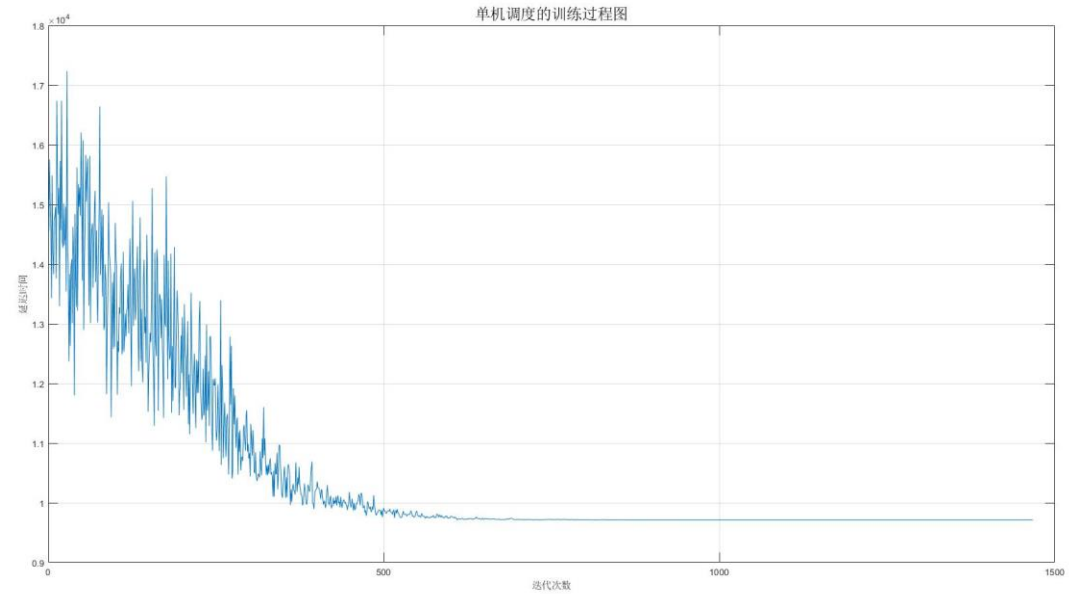

图 10：样例五的延迟时间变化

从图中可以看出，模拟退火算法会随着迭代次数的增加，温度逐渐下降的过程中，得到的代价值会上下波动，但大致趋势是越来越小，并逐渐稳定在某一阈值内。这说明该算法会随机跳出当前区域的局部最优解，表明了模拟退火算法应用于单机调度中的可行性。

§5 总结评价

为了解决生产生活中经常碰到的单机调度问题，本文考虑了局部搜索的改进算法——模拟退火。在建立的目标规划模型中，我们希望为工件分配的调度总延迟最小。模拟退火算法充分利用了其局部串行搜索能力强的特点，能够更有效、收敛到全局最优解。

但是，在调度总延迟的优化过程中，模拟退火算法经过了经过 400~500 次迭代，才得到比较稳定的数值。所以，我们可以适当利用自适应遗传算法全局并行搜索能力强的特点，将其与模拟退火算法结合，得到更加快速的得到全局最优解的“局部搜索”算法。

§6 参考文献

[1] 殷守林，刘天华，李航．基于模拟退火算法的卡尔曼滤波在室内定位中的应用研究[J]．沈阳师范大学学报（自然科学版）,2015,(1):86-90.doi:10.3969/j.issn.1673-5862.2015.01.019.

[2] 吴成明,王毅,毕红续,等．基于不同条件的旅游路线规划问题研究[J]．数学的实践与认识,2016,(15):90-96.

[3] 俞庆生,林冬梅,王东．多旅行商问题研究综述[J]．价值工程,2012,(2):166-168.doi:10.3969/j.issn.1006-4311.2012.02.095.

[4] 庞峰．模拟退火算法的原理及算法在优化问题上的应用[D]．2006.

[5] 陈华根,李丽华,许惠平,等．改进的非常快速模拟退火算法[J]．同济大学学报（自然科学版）,2006,(8):1121-1125.doi:10.3321/j.issn:0253-374X.2006.08.027.