最优化算法解决流水调度问题

尉坤（学号：1320221091）

摘　要：

本文主要解决流水调度问题，在满足一个机器同时加工一个工件，一个工件同时被一个机器加工，机器的加工顺序相同，只能从第一台到最后一台的条件下，求工件的加工顺序，使得得到最小的总完工时间。我利用 matlab 编写模拟退火算法与遗传算法解决此问题。模拟退火算法中通过随机交换两个工件的顺序，比较交换前后的总完工时间来确定是否交换。遗传算法中利用基因选择，基因交叉，基因突变，种群更新解决此问题。对于文件中的 11 个测试用例，模拟退火完成的时间通常短于遗传算法，结果的稳定性较高，优于遗传算法。两种算法的总完工时间结果相近，调度顺序有所不同。

关键词：

模拟退火算法，遗传算法，流水调度问题

1　引言

流水车间调度问题是指在具有多个工作站（机器）的流水线车间中，如何合理安排生产任务的问题，以最小化完成所有任务所需的总时间或最大化生产效率。该问题通常涉及到工件的顺序、各个工作站的处理时间以及每个工作站的容量等因素。需要用 m 台机器加工 n 个工件，每个机器只能同时加工一个工件，一个工件只能在一台机器上加工，每个机器的加工顺序相同（从第一台机器到最后一台机器）。求 n 个工件的加工顺序，使得最小化总完工时间。

Pij 为工件 i 在 k 时刻被加工的机器数

Pjk 为机器 j 在 k 时刻加工的工件数

Xij 为工件 i 在第 j 台机器上的顺序

Cij 为工件 i 在机器 j 上的加工时间

$$\min \ z = \sum_{j=1}^{n} \sum_{i=1}^{m} Cij \tag{1}$$

$$\text{s.t.}\ \forall \text{i, j, k} > 0$$

$$\sum_{j=1}^{m} Xij = 1 \tag{2}$$

$$\sum_{i=1}^{m} Xij = 1 \tag{3}$$

$$Pjk = 1 \tag{4}$$

$$Pik = 1 \tag{5}$$

$$\text{var Pij >= 0 Pjk >= 0 Xij >= 0 Cij >= 0}$$

(1)为目标函数，(2)表示每个工件加工顺序相同，即从第一台到最后一台。(3)表示每台机器加工工件的顺序相同，第二个工件不能先于第一个工件加工。(4)表示一台机器只能同时加工一个工件。(5)表示一个工件只能同时被一台机器加工。

定义了 4 个决策变量，通过对决策变量的约束使得满足每个工件被加工顺序相同，每台机器加工工件顺序相同，一台机器同时只能加工一个工件以及一个工件只能同时被一台机器加工的条件。目标函数为所有工件在所有机器上的完成时间之和最短。

主要运用 matlab 进行模拟退火算法和遗传算法的编写解决此流水车间调度问题。模拟退火算法中先对固定工件顺序的时间函数进行编写，之后随机交换两个工件的加工顺序计算总加工时间与原加工时间比较，保留最短时间，同时记录下工件的加工顺序。进行降温循环，得到最终结果。遗传算法中采用基因选择、基因交叉、基因突变产生新的优化个体，同时更新种群，达到循环次数后得到最优解。

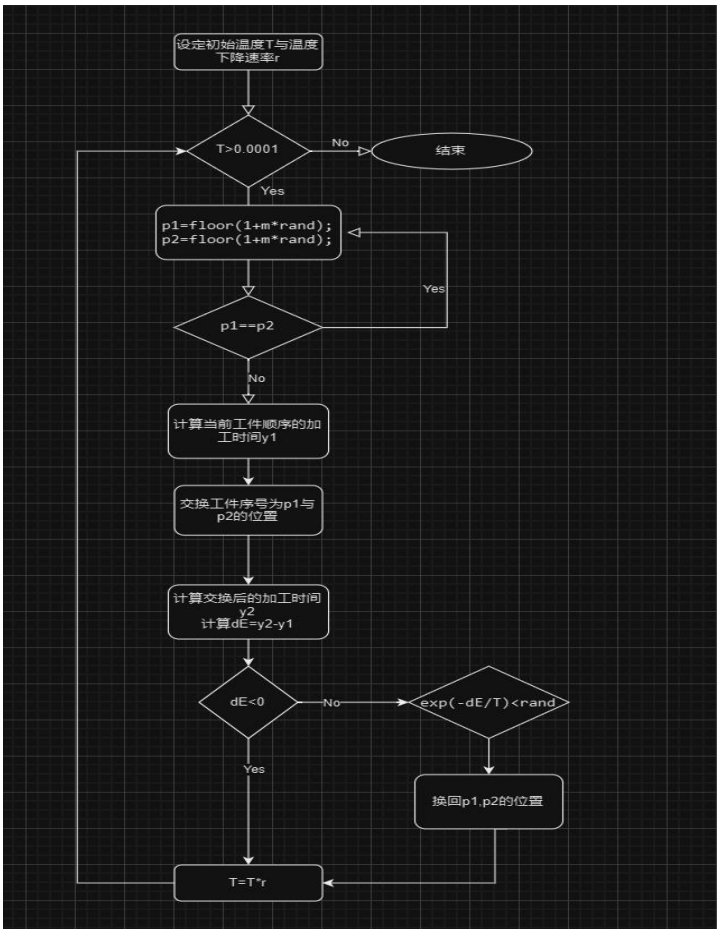
2 算法设计

2.1 模拟退火算法

模拟退火算法从某一较高的初始温度开始，随着温度参数的不断下降，结合概率突跳特性在解空间中随机寻找目标函数的全局最优解，即在局部最优解位置时，该算法能概率性地跳出局部最优解，并最终趋向于全局最优解的位置。模拟退火算法是通过赋予搜索过程随时间变化且最终趋于零的概率突跳性，从而可以有效地避免在求解过程陷入局部极值，并且最终能够趋于全局最优的串行结构的优化算法[1]。

模拟退火算法通过随机性搜索和接受劣解的方法得到最优解，模拟固体退火的过程。包括初始解生成、定义能量函数、温度设置、状态转移、接受劣解、温度更新与终止条件几部分。程序中通过设定温度初始值 T 与温度下降率 r，每进行一次循环，温度按照 T=T*r 下降，直到 T<0.0001 时循环结束，在循环中，主要通过 rand 产生的随机工件序号交换顺序，将交换之前与交换之后的加工时间进行比较，如果时间变短，则继续

循环，如果完成时间没有变短，则有 1-exp(-dE/T)的概率将两工件顺序换回。下图为模拟退火算法的流程图。



通过随机交换工件位置可以减少陷入局部最优的情况，更好地进行全局搜索。

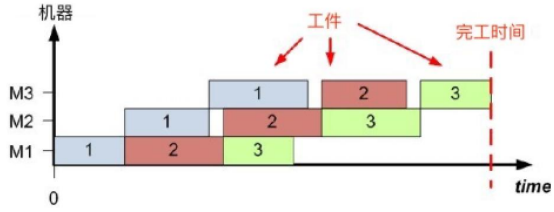
对于加工完成时间，设定 pretime 表示上一个工件在机器 X 中完成的时间，保证机器空闲。设定 curtime 表示当前工件在机器 X-1 中完成的时间，保证工件空闲，temp 为给出的测试用例，m 为工件数，n 为机器数。首先对 pretime 与 curtime 进行初始化，如下图例子所示，pretime 为 5，5+9=14，5+9+5=19，curtime 为 5，5+7=12，5+7+9=21。之后从 2 开始循环，需要同时在工件空闲与机器空闲的条件下才能进行下一工件的加工。需要比较 pretime 与 curtime 的时间，选取最长时间进行总时间的计算。不断更新 pretime 与 curtime 的时间值，对每个机器和工件都遍历计算，最终 pretime(n)为总加工时间，此为固定顺序下的工件加工总时间。

问题描述-例子

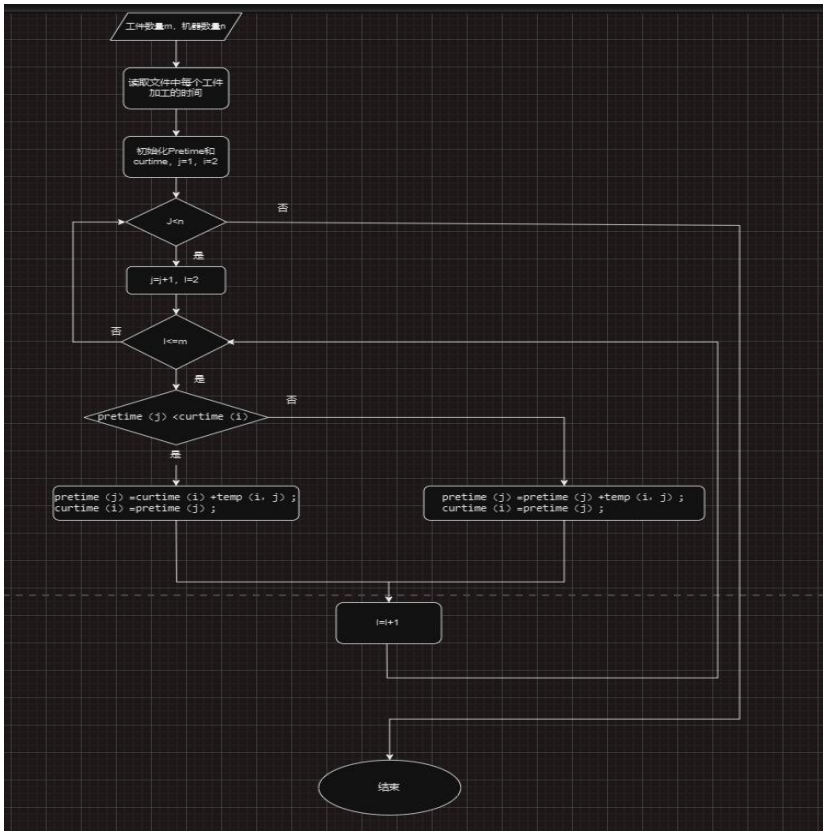
- 3台机器
- 3个工件
- 加工时间表

工件	机器1	机器2	机器3
工件1	5	7	9
工件2	9	9	7
工件3	5	9	5

调度方案为1→2→3时的加工情况（甘特图）



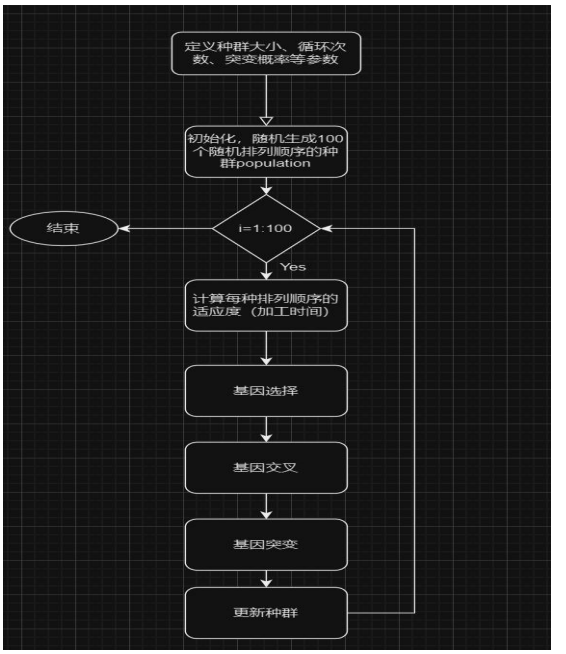
绘制加工完成时间流程图如下图。



2.2 遗传算法

遗传算法是一种启发式搜索和优化技术，核心思想是模拟自然界中生物进化的过程，主要模拟基因选择，基因交叉与基因突变。主要经过初始化种群、适应性评估、选择、交叉、变异、更新种群和终止条件检查几

个步骤。下图为遗传算法的流程图。



基因选择函数中将计算出的适应度进行排序，找到加工时间最短的两个工件排列顺序为父代。之后进行基因交叉，参考旅行商问题的基因交叉，利用随机数函数随机选取一段序列，以父序列为例，从选取序列后一个基因开始，比较母序列是否与被截取的基因片段重复，若不重复，接着写入父序列，若重复，跳到下一个基因值。最终得到两个交叉后的子代。基因有一定概率产生突变，通过设置突变概率 mrate 的值进行控制。基因突变采取随机交换两个基因的顺序进行实现。在更新种群部分，我对 population 中所有排列顺序进行适应度计算，同时对适应度排序，用得到的子代替换排在最后（加工时间最长）的两个个体，达到更新种群的目的。

主函数中进行种群初始化，在循环中计算适应度，基因选择，基因交叉，基因突变等操作，记录最小加工时间并不断更新，最终得到最优解。

3 实验

3.1 实验设置

本实验在 matlab 中进行，模拟退火实验代码在 sa.m 中，遗传算法代码在 ga.m 中，11 个测试用例保存在 example_1-example-11 中。

只需更改工件数 m，机器数 n，mat 文件下载的路径即可。如下图所示。

```
%输入
m=11;%工件的数量
n=5;%机器的数量
temp = load('F:\matlab xiazai\optimization\example_1.mat').example_1;
```

在模拟退火实验中，设定的参数有初始温度 T，温度下降率 r，分别用 index 与 Y 记录最优的工件排列顺序与每次筛选的工件加工时间，可根据 Y 画出迭代次数与最优值的图像。

```
index=1:m;
T=1000;%温度的初始值
r=0.98;%控制T下降的快慢
```

在遗传算法实验中，设定的参数有种群规模 pop，循环次数 generation，突变概率 mrate，初始种群 population。用 F 与 resume 分别记录迭代的加工时间与工件顺序，可根据 F 画出迭代次数与最优值的图像。

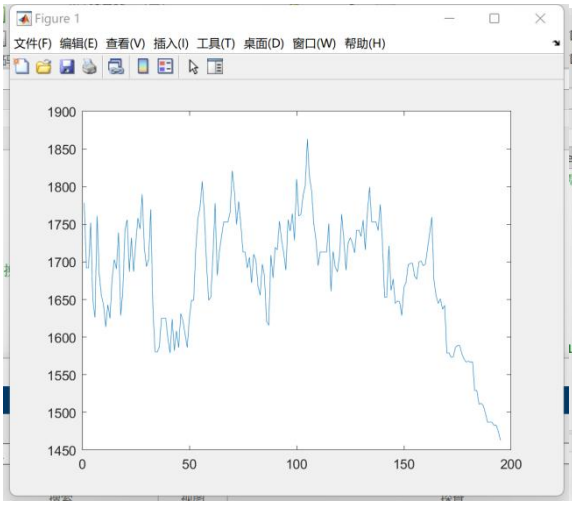
```
pop=100;%定义种群大小
generation=100;%循环次数
mrate=0.1;%突变概率
```

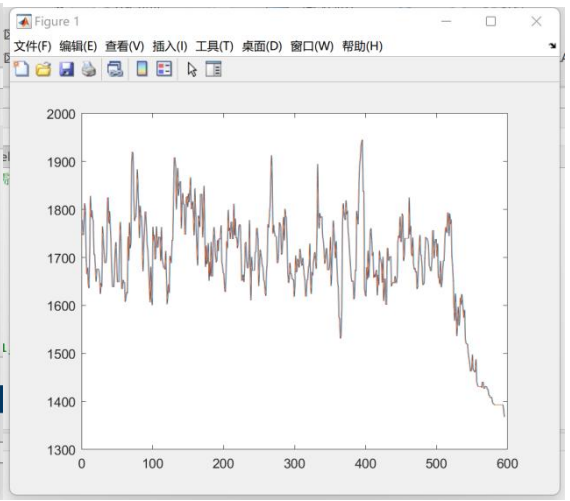
```
%初始化
population=zeros(pop,m);
for i=1:pop
    population(i,:)=randperm(m);
end
sign=1;
F=[];%记录每次迭代最小的适应度
resume=[];%记录每次迭代的工件排列顺序
```

可以更改这些实验参数进而达到更好的迭代效果。模拟退火与遗传算法的运行时间大概在 1s 以内，遗传算法的运行时间通常长于模拟退火的运行时间。

3.2 模拟退火算法参数实验

对第 7 个示例进行实验，因为第 7 个例子有 18 行 10 列，数据量较大，效果较明显，其他多数示例在 T=1000，r=0.98 的情况下可得最优解，并有一定的稳定性。



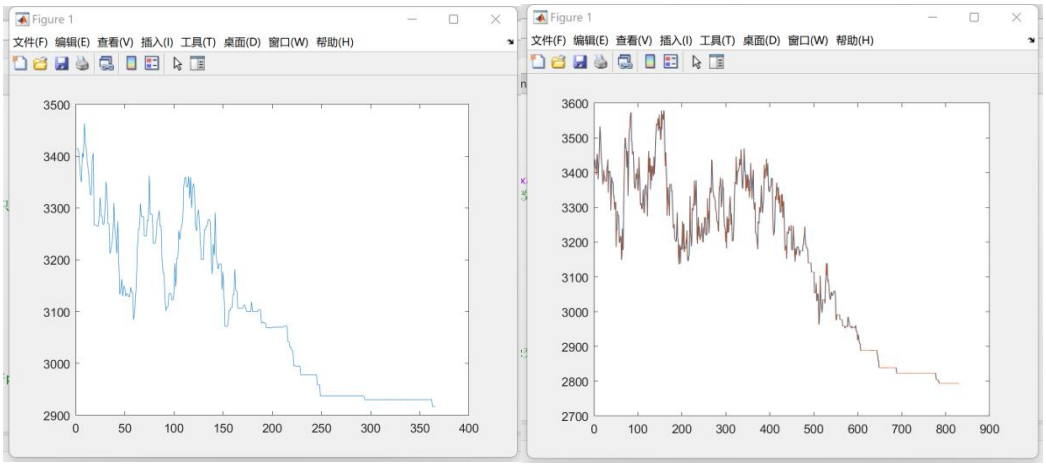


左图为初始温度 T=1000，r=0.98 中迭代次数与最优解的关系，右图为初始温度 T=10000，r=0.99 中迭代次数与最优解的关系。可以看出升高初始温度，减少初始温度下降的速率可以找到更小的最优解，循环次数显著增加。但也增加了最优解的随机性与不稳定性，每次循环后得到的解变化较大，结果不相同，需要多次运行找到最低值。因此应适当调整初始温度与下降速率的数值，合理确定循环次数。

通过改变初始温度与下降速率进行初步尝试，结果如图所示。

初始温度 T	降温速率 r	最优解	排列顺序
1000	0.98	1463	13 6 18 9 10 14 8 3 17 12 15 11 4 2 7 5 1 16
1000	0.99	1388	13 12 9 2 10 8 15 14 1 11 4 17 18 7 3 6 16
10000	0.98	1416	15 12 9 8 11 14 2 7 6 10 13 16 4 3 1 17 18 5
10000	0.99	1368	15 2 13 8 11 10 18 12 17 3 1 14 9 4 7 6 16 5

对第 11 个例子进行了实验，其中 m=40，n=10，左图为初始温度 T=1000，r=0.98 中迭代次数与最优解的关系，右图为初始温度 T=10000，r=0.99 中迭代次数与最优解的关系。



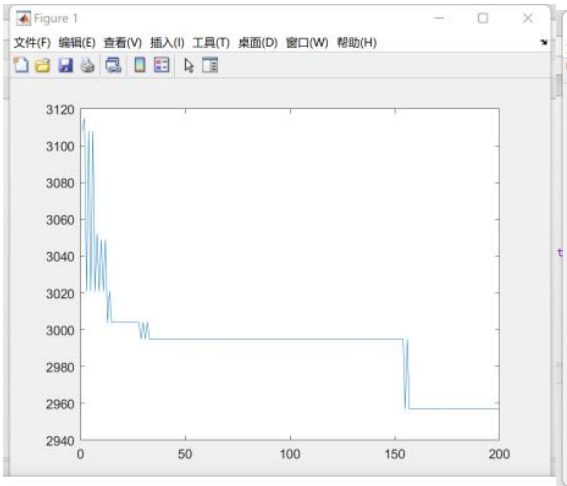
下面为例 11 的实验表格

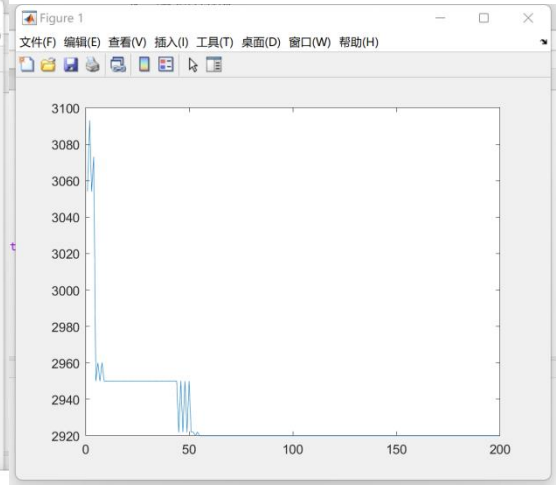
初始温度 T	降温速率 r	最优解
1000	0.98	2917
1000	0.99	2809
10000	0.98	2839
10000	0.99	2789

同时对其他示例进行实验发现 T=1000，r=0.98 可以达到最优解的要求，结果稳定，进而大部分可以采用此参数，但面对数据量较大的数据，可以适当提高初始温度 T 与减少下降速率从而找到更好的最优解。

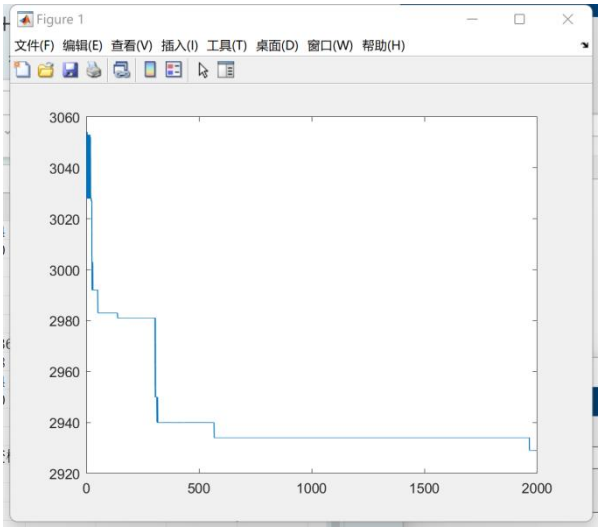
3.3 遗传算法参数实验

遗传算法中的实验参数包括种群大小，循环次数与突变概率，考虑到数据规模问题，第 11 个例子数据较丰富，下图是对例 11 的测试与分析。





左图为突变概率 mrate=0.1 时迭代关系与最优解的关系，右图为突变概率 mrate=0.3 时迭代关系与最优解的关系。其中种群规模与循环次数都为 100.



上图为将种群规模与循环次数设为 1000，突变概率为 0.1 时的每次迭代最优解。

种群大小 pop	循环次数 generation	突变概率 mrate	运行时间(s)	最优解
100	100	0.1	1.061	2957
100	100	0.3	0.898	2893
1000	1000	0.1	77.247	2929
1000	100	0.1	7.578	2902
100	1000	0.1	7.707	2883

100	1000	0.3	7.618	2856
500	500	0.3	19.553	2858
500	100	0.3	3.991	2946
100	500	0.3	4.000	2967

经过不断尝试改变参数，测试了运行时间与最终解的值，结果如表中所示。与模拟退火相比，在增加循环次数，种群规模时，遗传算法的运行时间显著增长，但适当地增加循环次数与增加突变概率可以找到更优的解。表中种群数量与循环次数为 1000 时，时间达到 77.247 秒，时间复杂度增长，由于算法的随机性可能运行一次得不到最好的解，大大增加了时间成本；当种群数量为 100，循环次数为 1000 或种群数量为 1000，循环次数为 100 时，运行时间大约为 7 秒，最优值较之前有所优化。将突变概率由 0.1 增长为 0.3 可以增加随机性，扩大搜索区域，更有可能找到更优的解。因此，需要综合考虑运行时间与最优解等因素，根据问题与实验数据的规模进行合理选择。

对于数据量比较小的问题，对本次实验的 11 个示例，种群数量与循环次数为 100，突变概率为 0.3（或 0.1）可以在很短的时间内找到较优的解。

3.4 算法对比

示例	算法	运行结果	运行时间（秒）
example_1	SA	7038	0.165
	GA	7038	0.353
example_2	SA	6269	0.274
	GA	6269	0.306
example_3	SA	5066	0.296
	GA	5071	0.347
example_4	SA	6666	0.313
	GA	6666	0.394
example_5	SA	9431	0.377
	GA	9581	0.429
example_6	SA	7044	0.311

	GA	7047	0.394
example_7	SA	1463	0.481
	GA	1478	0.628
example_8	SA	1941	0.390
	GA	1940	0.700
example_9	SA	975	0.278
	GA	1000	0.520
example_10	SA	1810	0.353
	GA	1857	0.673
example_11	SA	2917	0.364
	GA	2957	1.061

通过比较发现模拟退火的完成时间相对快一些，同时可以得到较优的最优解，但两者差异不大，都可以实现流水调度问题的解决。同时由于用到随机数，每次运行时的最优解会有变化。

最优调度方案

以下是针对两种算法的结果与调度顺序得出的最有调度方案。

示例	最优调度方案	加工完成时间
example_1	8 5 3 11 7 9 4 1 10 6 2	7038
example_2	4 2 1 5 3	6269
example_3	4 8 6 5 1 2 3 7	5066
example_4	7 5 9 1 6 8 10 4 3 2	6666
example_5	3 13 6 9 8 11 5 14 12 1 7 2 15 4 10	9431
example_6	3 4 1 5 8 2 7 6	7044
example_7	13 6 18 9 10 14 8 3 17 12 15 11 4 2 7 5 1 16	1463
example_8	11 17 7 5 13 14 15 4 12 18 16 2 10 1 8 9 3 6	1940
example_9	5 4 13 15 8 12 3 1 7 2 10 6 9 14 11 16	975
example_10	11 10 3 9 14 2 15 13 1 16 12 8 5 7 6 4	1810
example_11	8 37 40 5 34 11 16 3 35 1 28 14 39 13 29 36 23	2917

	15 22 31 7 10 9 27 30 2 38 4 17 19 20 32 12 18 21 24 33 6 25 26	

3.5 算法改进

通过资料查询与结合问题的思考，总结出几点可以改善算法效率的策略与思路。优化算法即提高算法的准确度与计算速度（收敛性）。对于遗传算法，应尽量采用浮点编码，可直接表征优化变量达到较高的计算精度。保证优化变量最少问题，减少优化变量数可以降低问题的复杂程度，算法容易收敛。同时考虑问题的约束条件与先验知识帮助求解。遗传算法中大群体量与少计算量要求，高变异概率与算法收敛性间都存在矛盾关系，参数的设置需要经过不断调试，根据群体的差异设置交叉概率与变异概率。可以根据算法间的优缺点运用混合遗传算法。第一种为增加局部搜索的混合遗传算法，用模拟退火算法进行局部搜索，弥补遗传算法跳解，局部搜索能力不足的缺点，将遗传算法与模拟退火算法相结合可能会达到更好地效果。第二种为优化部分关键代码，缓解遗传算法的计算时间过长，全局提高收敛能力[2]。遗传算法的种群初始化中可以先随机生成一半的初始种群，之后将一般种群进行逆序排列，从而有效避免初始种群聚集。在变异操作中选取 2 个作业数进行变异，增大变异程度。同时还可以进行多目标选择，高交叉概率，低变异概率与低交叉概率、高变异概率的相互切换[3]，帮助在较大范围与较短时间找到最优解。

4 总结

本文主要利用 matlab 编写模拟退火与遗传算法解决流水车间调度问题，首先对问题进行数学建模，分析出目标函数，决策变量与约束条件，对题目进行深入的理解。接着，详述了模拟退火算法与遗传算法的算法思想和关键步骤，对解决此问题采用的思路与方法，用流程图进行更直观的体现。之后，还原实验条件，保证算法的可操作性，分析算法中实验参数的影响，对更合适的实验参数的选取进行尝试，对比了不同参数下的最优解与运行时间等，针对不同的问题可以选取不同的实验参数。最后，对文件中 11 个示例进行测试，得出结果，以表格的方式呈现出最短加工时间与最有调度顺序，进行调度问题的解答。

本文中算法的实现主要由作者手动模拟与编写，可能需要进一步验证。只是基本上解决了调度问题，并为对代码进行进一步优化与精炼，同时还需阅读大量资料，学习其他人更好的解决思路，将所学融入到本文算法中，从而可以更高效更准确地解决问题得到最优解。同时，由于采用随机函数，算法的不确定性较高，每次运行时的结果可能不太一样，需要多次运行，找到更优的结果。

参考文献:

[1] 郭林陇，于大国，李永杰，陈路生，杜慧福.基于模拟退火算法的空间直线度误差评定[J].工具技术. 2024,58(05).

[2] 刘攀，郭生练，李玮，易松松.遗传算法在水库调度中的应用综述[J].水利水电科技进展. 2006(04).

[3] 徐嘉琦，田野.基于改进遗传算法的柔性流水车间调度研究[J].制造技术与机床. 2024(04).