首先程序输出的结果如下：

Simulated Anneal(day 1):

Solution : [0, 1, 0, 1]

Utility of solution: 0.7666666666666667

Simulated Anneal(day 2):

Solution : [1, 1, 1, 1]

Utility of solution: 0.6605555555555556

Simulated Anneal(day 3):

Solution : [0, 0, 1, 0]

Utility of solution: 0.5266666666666667

Simulated Anneal(day 4):

Solution : [1, 1, 0, 1]

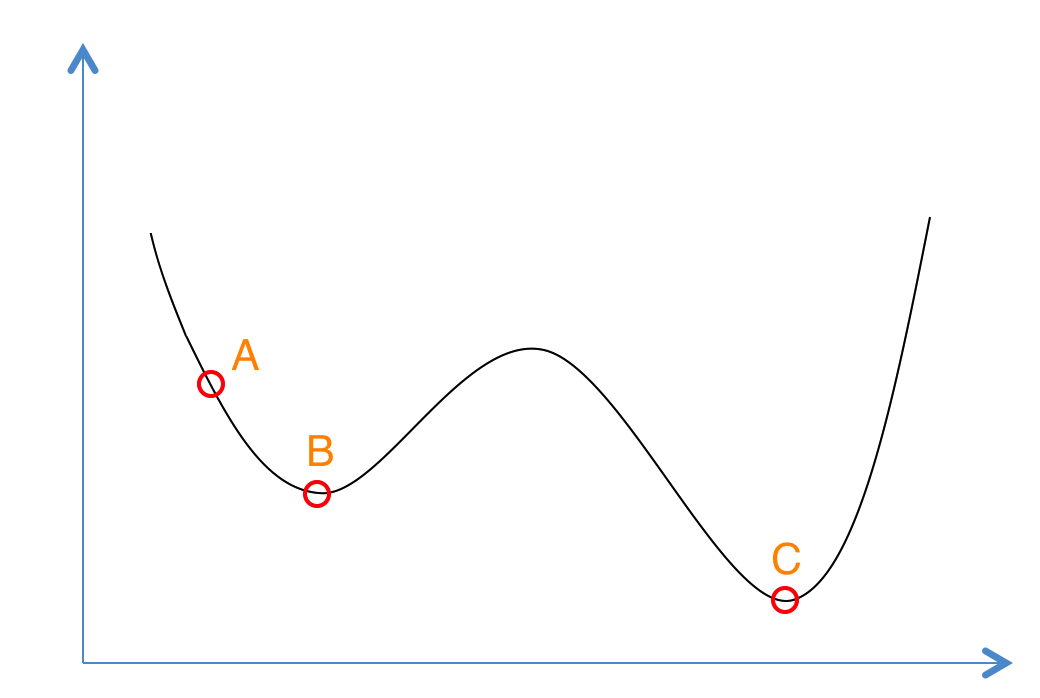
Utility of solution: 0.88

可以看出，输出的是与文献一样的结果。

1. 模拟退火算法原理：

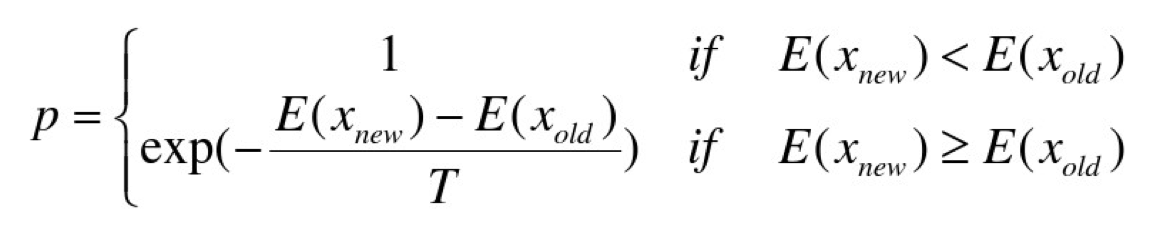
在热力学上，退火（annealing）现象指物体逐渐降温的物理现象，温度愈低，物体的能量状态会低；够低后，液体开始冷凝与结晶，在结晶状态时，系统的能量状态最低。大自然在缓慢降温（亦即，退火）时，可“找到”最低能量状态：结晶。但是，如果过程过急过快，快速降温（亦称「淬炼」，quenching）时，会导致不是最低能态的非晶形。

模拟退火其实也是一种Greedy算法，但是它的搜索过程引入了随机因素。模拟退火算法以一定的概率来接受一个比当前解要差的解，因此有可能会跳出这个局部的最优解，达到全局的最优解。



以上图为例，模拟退火算法在搜索到局部最优解B后，会以一定的概率接受向右继续移动。也许经过几次这样的不是局部最优的移动后会到达B 和C之间的峰点，于是就跳出了局部最小值B。

根据Metropolis准则，粒子在温度T时趋于平衡的概率为exp(-ΔE/(kT))，其中E为温度T时的内能，ΔE为其改变数,k为Boltzmann常数。Metropolis准则常表示为：



我们将一次向较差解的移动看做一次温度跳变过程，我们以概率P(dE)来接受这样的移动。也就是说，在用固体退火模拟组合优化问题，将内能E模拟为目标函数值f，温度T演化成控制参数t，即得到解组合优化问题的模拟退火演算法：由初始解i和控制参数初值t开始，对当前解重复“产生新解→计算目标函数差→接受或丢弃”的迭代，并逐步衰减t值，算法终止时的当前解即为所得近似最优解，这是基于蒙特卡罗迭代求解法的一种启发式随机搜索过程。退火过程由冷却进度表(Cooling Schedule)控制，包括控制参数的初值t及其衰减因子Δt、每个t值时的迭代次数L和停止条件S。

总结起来就是：

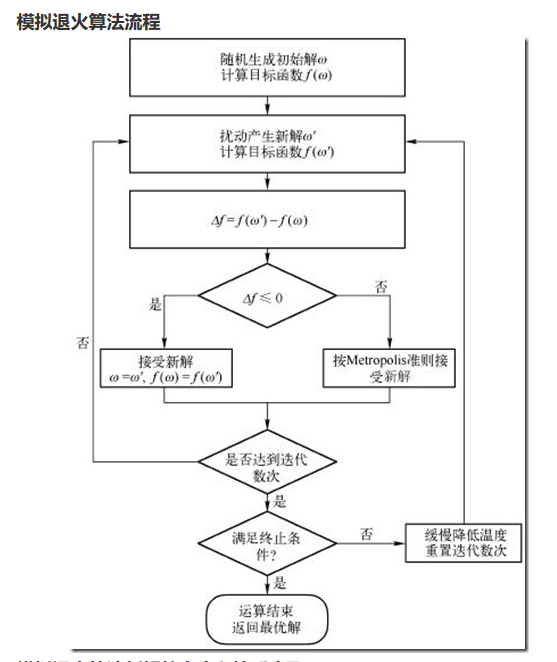
若f(Y(i+1))<=f(Y(i))(即移动后得到更优解)，则总是接受该移动；

若f(Y(i+1))>f(Y(i))(即移动后的解比当前解要差)，则以一定的概率接受移动，而且这个概率随着时间推移逐渐降低（逐渐降低才能趋向稳定）相当于上图中，从B移向BC之间的小波峰时，每次右移（即接受一个更糟糕值）的概率在逐渐降低。如果这个坡特别长，那么很有可能最终我们并不会翻过这个坡。如果它不太长，这很有可能会翻过它，这取决于衰减t值的设定。

模拟退火算法计算标准步骤：

1. 确定要优化的目标函数
2. 初始化算法相关参数（起始温度、降温系数、停止迭代的温度阈值、搜索步长，最大迭代次数等）
3. 确定迭代循环终止条件（终止温度、最大迭代次数、系统熵稳定情况等，目标函数值迭代情况等）
4. 确定状态转移方程（其实也就是产生新解的方法）。

在上述准备工作做好后，模拟退火算法具体的计算流程其实很简单：



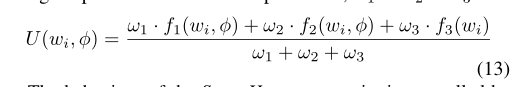
1. 本项目中模拟退火算法的应用

在这个项目中，其实最关键的是解空间的表示以及新解的产生方法：

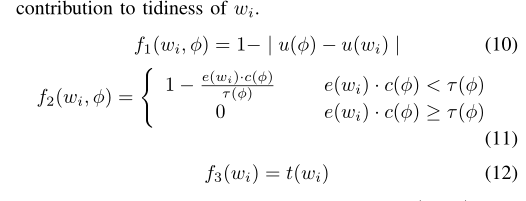
首先对于工作流解的表示，我这里采用和文献中一样的方法，即采用布尔编码:[1,0,1,1,...,0],里面每一位数代表一个设备（与DeviceInfo.csv中的设备从上到下一一对应，1代表这个设备被选在工作流中，0代表不被选取）。

其次对于新解的产生方法：一般来说，为便于后续的计算和接受，减少算法耗时，通常选择由当前新解经过简单地变换即可产生新解的方法，如对构成新解的全部或部分元素进行置换、互换等，我这里采用的就是随机对设备编码表（如[1,0,1,1,...,0]）中部分元素进行变换：如果原来0就变成1，如果原来是1就变成0，具体要变换的元素的个数由“搜索步长”这一外部参数给定。

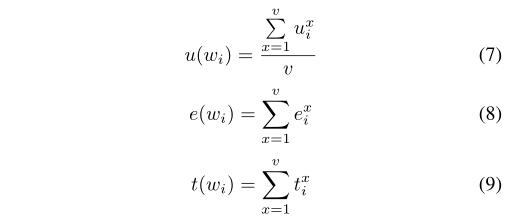
除此之外，本项目中要优化的目标函数就是文献中的Overall Utility函数：



其中f1,f2,f3如下所示：



其中u(wi),e(wi),t(wi)函数如下所示：



这些公式和流程的具体实现请看代码，注释比较详细。