贪心算法例题分析

赵耀

应用贪心算法注意2点

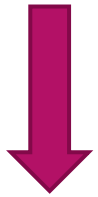
- 该用哪种规则去选择？跟需求和约束强相关
- 如何证明你所设计的规则是可以得到最优解的

任务调度问题1

- 设有n个任务同时等候操作系统调度。该系统只有1个CPU。假设每个任务的执行时间为t[i], 1≤i≤n。应如何安排n个任务的调度顺序，使得平均等待时间达到最小？平均等待时间是n个任务等待CPU执行的时间总合除以n。

任务调度问题1

- 设有n个任务同时等候操作系统调度。该系统只有1个CPU。假设每个任务的执行时间为t[i], 1≤i≤n。应如何安排n个任务的调度顺序，使得平均等待时间达到最小？平均等待时间是n个任务等待CPU执行的时间总合除以n。



直觉上每次应该选择执行时间最短的任务

执行时间的计算

假设Wi为排序后排在i的任务执行时间

第一个任务的等待时间：0

第二个任务的等待时间：W[1]

第三个任务的等待时间：W[1]+W[2]

……

第n个任务的等待时间：W[1]+W[2]+…W[n-1]

平均等待时间 = ((n-1)W[1]+(n-2)W[2]+…+1*W[n-1])/n

要使等到时间最短，由上式可知，系数最大时，W值应最小

用交换证明的过程：
比较贪心算法的解 G=j1, j2, …, jr, jr+1…
和最优解S=j1, j2, …, jr, j'r+1, …., jr+1, …
假设前r项贪心的选择和最优解的选择是一样的，从第r+1项开始不同。
将最优解的j'r+1和jr+1交换，我们得到新的解S'=j1, j2, …, jr, jr+1, …., j'r+1, …
显然S'中j1, j2, …, jr的等待时间与S一样，j'r+1之后的等待时间也与S一样，由于jr+1是jr+1, …., j'r+1段中最小的值， jr+1≤ j'r+1，所以jr+1, …., j'r+1段的等待时间也不会增加，所以S'也是一个最优解。由于S是最优解，所以jr+1只可能等于 j'r+1，贪心解与最优解在r+1处也是相同的选择。
重复这个过程，可以推断出最优解与贪心解是相同的。

任务调度问题2

- 设有n个任务同时等候操作系统调度。该系统有s个CPU。假设每个任务的执行时间为t[i], 1≤i≤n。应如何安排n个任务的调度顺序，使得平均等待时间达到最小？平均等待时间是n个任务等待CPU执行的时间总合除以n。

当n%s = 0

第一个任务的等待时间：0
第二个任务的等待时间：0
……
第s个任务的等待时间：0

第s+1个任务的等待时间：t[1]
第s+2个任务的等待时间：t[2]
……
第2s个任务的等待时间：t[s]

第2s+1个任务的等待时间：t[1]+t[s+1]
第2s+2个任务的等待时间：t[2] +t[s+2]
……
第3s个任务的等待时间：t[s] +t[2s]

第3s+1个任务的等待时间：t[1]+t[s+1]+t[2s+1]
第3s+2个任务的等待时间：t[2] +t[s+2] +t[2s+2]
……
第4s个任务的等待时间：t[s] +t[2s] +t[3s]

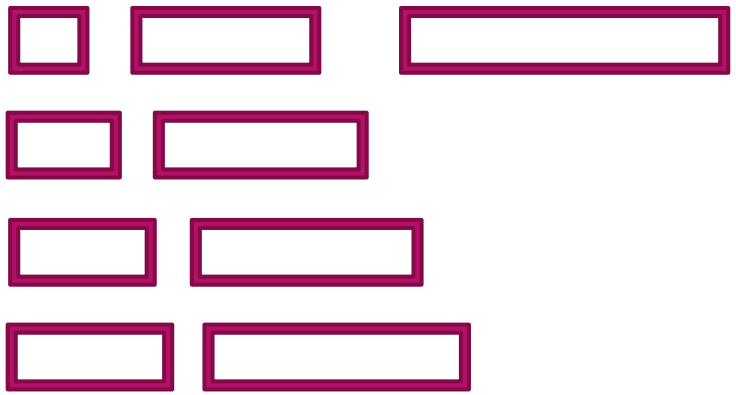
平均等待时间 = ((n/s-1)(t[1]+ t[2]+... t[s])+(n/s-2)(t[s+1]+t[s+2]+...+t[2s])...+1*(t[n-s+1]+...+t[n]))/n
证明过程，与前类似。

结论

- 当n%s = 0 时，最优解满足前s个第一批分配给s个cpu，前s+1到前2s个第二批分配，依次类推。贪心选出的解为最优解之一

当n%s != 0

- 假设n%s=k, 可推出 (n-k)%s = 0 ， 由上述可知， 前n-k个任务由贪心算法可以得到最优分配，而非贪心算法在前n-k个的分配上与贪心算法只有同批次任务在s的执行排列不同
- 用贪心算法根据t[i]分配剩下的k个任务，假设存在一种最优解分配k个任务能得到最短等候时长。通过整除的分析可知，最优解与贪心解的区别在于同批次的执行排列不同，可以理解为贪心解通过多次任务交换可以与最优解一致。下述证明，最优解按照贪心解来交换，每一次交换后的解不会比原来的解差。
- 同行交换，t[i]<t[j] 大值前调，调整后执行时间差值，2*t[j]+t[i]-(2*t[i]+t[j]) = t[j]-t[i]>0
- 同列交换, 每个s的任务数相同的交换，交换后结果相等，任务数不等的交换， t[i]<t[j] 大值上调，调整后2*t[j]+t[i]-(2*t[i]+t[j]) = t[j]-t[i]>0
- 不同列不同行交换，同理可证
- 综上，最优算法在每一次交换后都不可能产生次优的解，所以贪心算法产生的解也是最优解。



任务调度问题3

- 一个单位时间任务是恰好需要一个单位时间完成的任务。给定一个单位时间任务的有限集S。关于S 的一个时间表用于描述S 中单位时间任务的执行次序。时间表中第1 个任务从时间0 开始执行直至时间1 结束，第2 个任务从时间1 开始执行至时间2 结束，…，第n个任务从时间n-1 开始执行直至时间n结束。具有截止时间和误时惩罚的单位时间任务时间表问题可描述如下：
  (1) n 个单位时间任务的集合S={1,2,…,n} ；
  (2) 任务i的截止时间d[i],1≤i≤n,1≤d[i]≤n，即要求任务i在时间d[i]之前结束；
  (3) 任务i 的误时惩罚1≤w[i]<m,1≤i≤n,即任务i 未在时间d[i]之前结束将招致w[i]的惩罚；若按时完成则无惩罚。
- 任务时间表问题要求确定S 的一个时间表（最优时间表）使得总误时惩罚达到最小。

任务调度问题1：answer

- 每次选择t[i]最小的任务

任务调度问题2：answer

- 每次选择t[i]最小的任务依次分配1到s的CPU上

任务调度问题3：answer

- 根据误时惩罚的大小进行排序，将误时惩罚大的放在前面。开一个数组作为时间槽，记录每个单位时间是否有任务安排。每一次都贪心的选择误时惩罚大的任务，尽量将任务安排在其截止时间完成，否则放在放在前一天，以此类推。若在截止时间前都有任务安排，则将该任务放到时间槽的最后。
- 初始S为所有任务，A为空。迭代从0到n，每一次迭代i都从S中移除d值在i或i之前的任务，将这些任务与集合A中的任务一起按照w从高到低排序，只保留前i个任务，更新集合A，i之后的任务放到集合F中，将集合A中的任务按照d值从小到大顺次分配时间槽，F中任务依次分配在剩余的空槽中。