基于多机调度问题的动态规划算法

张凤梅[1],洪运国[2]
(1.辽宁师范大学 计算机与信息技术学院,辽宁 大连 116029;
2.大连职业技术学院 信息技术系,辽宁 大连 116035)

摘 要:动态规划设计策略对许多具有最优解的实际应用问题的解决是灵活和有效的。文中首先针对在多机系统的操作系统的一类多机调度问题进行了分析,并给出了该类问题的动态规划算法,最后对所给算法的复杂度进行了分析和讨论。实验结果验证了所提出方法的有效性。

关键词:动态规划;最优解;多机调度问题;复杂度
中图分类号:TP301.6 **文献标识码**:A **文章编号**:1005-3751(2006)03-0061-02

Dynamic Programming Algorithm for a Kind of Scheduling Problem of Multicomputer

ZHANG Feng-mei[1], HONG Yun-guo[2]
(1. College of Computer and Information Technology, Liaoning Normal University, Dalian 116029, China;
2. Department of Information and Technology, Dalian Vocational Technology College, Dalian 116035, China)

Abstract: The dynamic programming algorithm is a flexible and high - efficient method to many problems which have the best method. In this paper, a kind of scheduling problem of multicomputer is brought up firstly. And then a novel algorithm for this problem based on dynamic programming is proposed. Finally, the complexity of the proposed algorithm is analyzed. Simulation results show it is effective.

Key words: dynamic programming; the best method; scheduling problem of multicomputer; complexity

0 引 言

动态规划设计策略最早由 Bellman[1]等人提出。Bellman认为利用最优性原理以及所获得的递推关系去求解最优决策序列可以使问题的计算量急剧下降。动态规划思想是将待求解问题分解成若干个子问题,先求解子问题,然后从这些子问题的解得到原问题的解。动态规划算法通常用于求解具有某种最优性质的问题。几年来,人们在这一领域进行了积极的研究,给出了很多具有重叠子问题特性的动态规划算法。

文中首先对在多机系统的操作系统中一类 NP 问题[2],即多机调度难解问题进行了分析和讨论,对该问题的最优子结构性质进行了证明,在此基础上给出了该问题的递归结构,同时实现了其动态规划算法,最后对所提出的算法的复杂度进行了分析和讨论。

1 问题的提出

在多机系统中设有两台处理机 A 和 B 来处理 n 个作业,设第 i 个作业交给机器A处理时需要时间 a_i,若由机器B来处理,则需要时间 b_i。由于各作业的特点和机器的性能关系,很可能对于某些 i 有 $a_i \geqslant b_i$,而对于某些 j,$j \neq i$ 有 $a_j < b_j$,既不能将一个作业分开由两台机器处理,也没有一台机器能同时处理 2 个作业。

问题[3]:设计一个算法,使得这两台机器 A 和 B 处理完这 n 个作业的时间最短(从任何一台机器开工到最后一台机器停工的总时间)。

2 问题的动态规划算法

2.1 问题的最优子结构性质

设全部作业的集合为 $N = \{1,2,\cdots,n\}$,作业 $p \in N$ 是机器A上第一个调度作业;作业 $q \in N$ 是在机器B上第一个调度作业,$S = \{N - p - q\}$ 是 N 的作业子集,设 π 是所给 N 个作业的最优调度,它所需的调度时间为 $T(N, t)$,N 个作业调度问题的最优值 $T(N,0)$ 在一般情况下,机器A开始加工 N 中某个作业 p 时,所需时间为 $t1$;机器B开始加工 N 中另一个作业 q 时,所需时间为 $t2$,对于其它任意作业 $m \in S$,要等待 $t = \min\{t1, t2\}$ 之后,才可在机器A或B上被调度。

设 π 是所给 N 个作业的最优调度,它所需加工时间为 $\min\{t1, t2\} + T'$,其中 T' 是调度完 p 和 q 之后,作业集 S

收稿日期:2005-06-07
基金项目:国家自然科学基金资助项目(60372071);辽宁省自然科学基金资助项目(20032125)
作者简介:张凤梅(1970—),女,辽宁本溪人,硕士,讲师,研究领域为操作系统,Linux,算法分析与设计。

所需的调度时间，$S = \{N - \pi(p,q)\}$，则 $T' = T(S, \min\{t1,t2\})$。

证明：由 T 的定义可知，

$$T' \geqslant T(S,\min\{t1,t2\}) \quad (1)$$

若 $T' > T(S,\min\{t1,t2\})$，设 π' 是作业集 S 在机器A或B上的等待时间为 $\min\{t1,t2\}$ 情况下最优调度，则 $\pi(p,q) + T(S,\min\{t1,t2\}) < \pi(p,q) + T'$，这与 π 是 N 的最优调度相矛盾，故

$$T' \leqslant T(S,\min\{t1,t2\}) \quad (2)$$

由式(1)和式(2)可得 $T' = T(S,\min\{t1,t2\})$，从而证明了该作业调度具有最优子结构性质。

2.2 问题的递归结构

由作业调度问题的最优子结构性质[4]可知，$T(N,0) = \min\{\min\{a_i,b_i\}, T(N-\{i\},\min\{T_1,T_2\})\}$ $1 \leqslant i \leqslant n$

其中 T_1 和 T_2 分别为前 $i-1$ 个作业在机器A和B上分别被调度所花时间。即：

$T(S,t) = \min\{\min\{a_i,b_i\} + T(S-\{i\}), \max\{t - \min\{T_1,T_2\}\}\}, i \in S$

其中 $\max\{t - \min\{T_1,T_2\}\}$ 为调度完作业 i 之后，剩下作业在机器上还需的时间。

按照上述递归式，可设计出流水作业在多机系统中的作业调度问题的动态规划算法，但通过对递归式深入分析，算法可进一步简化。

2.3 算法的实现

设流水作业的个数为 N，用含有两个成员的结构数组 $a[N]$ 和 $b[N]$ 来存储 N 个作业在机器A和B上加工所需的时间和作业号，结构数组定义：struct AB { int time; int JobNum;}$a[N],b[N]$；用 min_ t 变量存放最后所需的最短时间，即最优值，用一个 int schedul[N] 来存储最后的最优解即所需时间最短的作业调度次序。

(1) 数组的初始化和一些变量的定义。

设 N 为6，假设6个作业分别在机器A和B上执行所需的时间分别为$(a1,a2,a3,a4,a5,a6) = (2,5,7,10,5,2)$；$(b1,b2,b3,b4,b5,b6) = (3,8,4,11,3,4)$；$a[N] = \{\{2,1\}, \{5,2\}, \{7,3\}, \{10,4\}, \{5,5\}, \{2,6\}\}$；$b[N] = \{\{3,1\}, \{8,2\}, \{4,3\}, \{11,4\}, \{3,5\}, \{4,6\}\}$；int $t1,t2,u,v$,min_ t,schedul[N]//min_ t 和 schedul[N] 分别是所求最优值和最优解；$t1,t2$ 是在A和B上调度分别所需时间；u,v 是调度指针。

(2) 最优值和最优解的算法描述。

① 对数组 a 和 b 分别按调度时间进行由小到大排序。

②求最优值和最优解，具体的算法如下：

```
void Bestvalue_method(struct AB a[],struct AB b[])
{ int n=N,m=0,min_t,t1,t2,
u,v,schedul[N];
if(a[u].time<b[v].time)
{ schedul[m]=a[u].JobNum;n--;t1+=a[u].time;
  query(b,schedul[m]);//在b中查找作业号=schedul[m]的b[j]将其time置为-1,表示相应的作业已在A上调度完成
  m++ ;u++;}
  else
  { schedul[m]=b[v].JobNum;n--;t2+=b[v].time;query(a,schedul[m]); //在a中查找作业号=schedul[m]的a[i]将其time置为-1,表示相应的作业已在A上调度完成
  m++;v++;}
  //经过一个判断进行第一次调度这样t1或t2有了一个非零的初值
 while (n>0)
{ if (t1>t2)
 {while(b[v].time==-1) v++;
 schedul[m]=b[v].JobNum;t2+=b[v].time;n--;
  query(a,schedul[m]);
  m++;v++
}
else
{ while(a[u].time==-1) u++;
 schedul[m]=a[u].JobNum,t1+=a[u].time;n--;
 query(b,schedul[m]);
 m++;v++;}}
min_t=max{t1,t2};}
```

//min_t 为A,B调度时间 t1 和 t2 最长的，min_t 即为所求的最优值，Schedul[N]即为N个作业在两台机器上在最优解

2.4 算法的结果和复杂性分析

应用上述算法，对 N 为6，假设6个作业分别在机器A和B上执行所需的时间分别为$(a1,a2,a3,a4,a5,a6) = (2,5,7,10,5,2)$；$(b1,b2,b3,b4,b5,b6) = (3,8,4,11,3,4)$，进行调度结果 min_ t 即所需的最短时间为18，schedul[N] 即最优解为{1,5,6,3,2,4}。

算法 BestValue_method 的主要计算量取决于程序对 n 和每次循环体内查询 query(a,schedul[m]) 或 query(b,schedul[m]) 计算量，循环体内计算量为 $O(1)$，而两次循环总次数为 $O(n^2)$（n 为流水作业的个数），算法所占用的空间为 $O(n)$[5]。

3 结束语

综合以上分析，整个多机调度问题的动态规划算法的时间复杂度为 $O(n^2)$，所占空间复杂度为 $O(n)$，这要比用其它算法（比如用分治法求解，其时间复杂度为指数级），效率较高。同时本算法可以稍加改动，推广到具有 $m > 2$ 个机器的 n 个流水作业的调度的问题上，同样有效。本算法主要应用在多机系统中，当操作系统有多个作业需要处理时，可以用此算法来预测调度次序，从而当有一个机器出现空闲时，按此算法预测出的调度次序进行调度，可以使这 m 个机器能在最短的时间内调度完这 n 个作业，提高计算机系统资源的利用率，加快用户的响应，从而使用户获得最合理的响应时间，满足不同用户的要求。

（下转第65页）

的极值点,而其位移则和 2^2 尺度上对应的极值点相同。

以上保留的极大值点的位置即可确定目标信号的精确位置。

4　模拟仿真与结论

图 1(a)为没有受干扰的真实目标信号,图 1(b)为受到低频干扰和白噪声污染的目标信号,信噪比为 - 6dB,采用文中小波变换的方法,利用 Matlab 软件[6],对图 1(b)中的信号进行三个尺度小波变换模极大值处理,如图 2 所示,图 1(c)是根据图 2 处理得到的目标定位脉冲。实验结果与真实目标信号的位置相符。

文中根据不同信号在相邻尺度间小波变换模极大值及传播特性不同,进行奇异目标信号的跟踪定位,具有很高的位置分辨率,获得了很好的效果,这是传统方法无法比拟的。

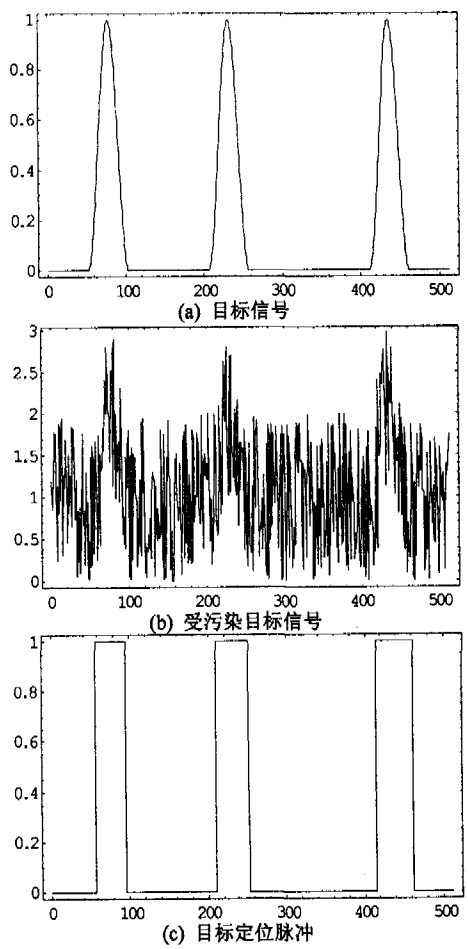

图 1　目标、污染和定位信号

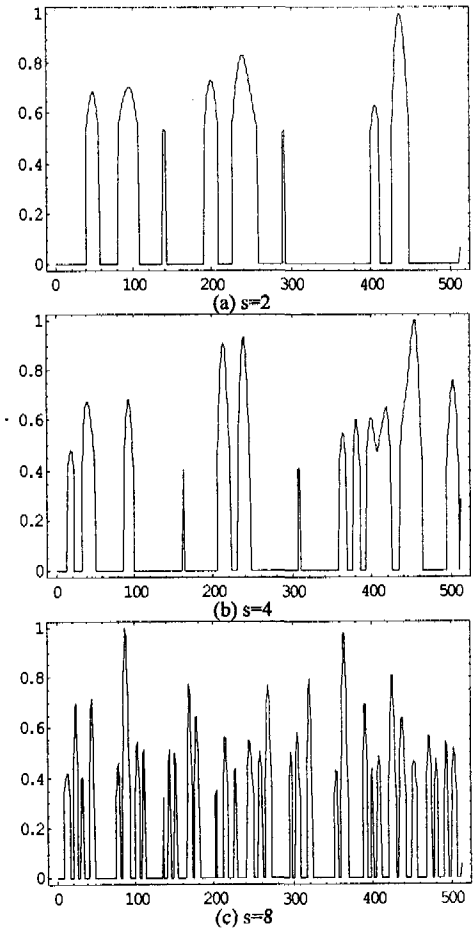

图 2　三个尺度下的小波变换

参考文献:

[1] 李世雄.小波变换及其应用[M].北京:高等教育出版社,1998.

[2] 秦前清,杨宗凯.实用小波分析[M].西安:西安电子科技大学出版社,1994.

[3] Mallat S. Singularity Detection and Processing with Wavelets [J]. IEEE Trans on Information Theory,1992,38(2):617 - 643.

[4] Mallat S, Zhong S. Characterization of Signal from Multiscale Edges[J]. IEEE Trans on Pattern Analysis and Machine Intelligence,1992,14(7):710 - 732.

[5] 王　俊,陈逢时,张守宏.一种利用子波变换多尺度分辨特性的信号消噪技术[J].信号处理,1996(2):105 - 109.

[6] 胡昌华,张军波.基于 MATLAB 的系统分析与设计——小波分析[M].西安:西安电子科技大学出版社,1999.

(上接第 62 页)

参考文献:

[1] Katoen J P. Dynamic programming algorithm[EB/OL]. http://www. aduni. org/courses/algorithms/ - handouts/Reciation-06. html,2001 - 02.

[2] 余祥宣,崔国华,邹海明.计算机算法基础[M].武汉:华中科技大学出版社,2000.

[3] 王晓东.计算机算法设计与分析[M].北京:电子工业出版社,2001.

[4] Dawes R. Thoughts on Dynamic programming [EB/OL]. http://www. cs. queensu. ca/home/cisc365 - /1998Dawes, 1998.

[5] 张南平,王海军.一种新型单循环排序算法[J].微机发展,2005,15(5):114 - 115.