**Report For Exercise Branch & Bound**

**Formal Lower bounds**

09019204

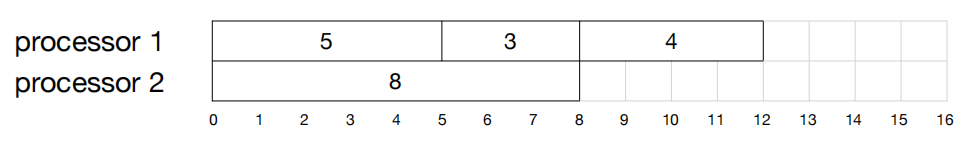
曹邹颖

1. **Problem description / demand analysis问题描述**

Use *LB* = max{*LB*1, *LB*2} for the design of a Branch & Bound algorithm

*E* = {5,3,8,4} and *m* = 2

Start from the following feasible solution: *x*1 = 1, *x*2 = 1, *x*3 = 0, *x*4 = 1



1. Complete the tree of the Branch & Bound, and find the optimal solution
2. Give the order of the nodes you explored
3. **Results and analysis结果和分析**

解：,

, ,

Where is the load of the current solution,

;

1. 本题从*x*1 = 1, *x*2 = 1, *x*3 = 0, *x*4 = 1该解开始，得到

*x*1 = 1时 the remaining jobs is{3,8,4},,

,

;

*x*1 = 1, *x*2 = 1时 the remaining jobs is{8,4},,

,

;

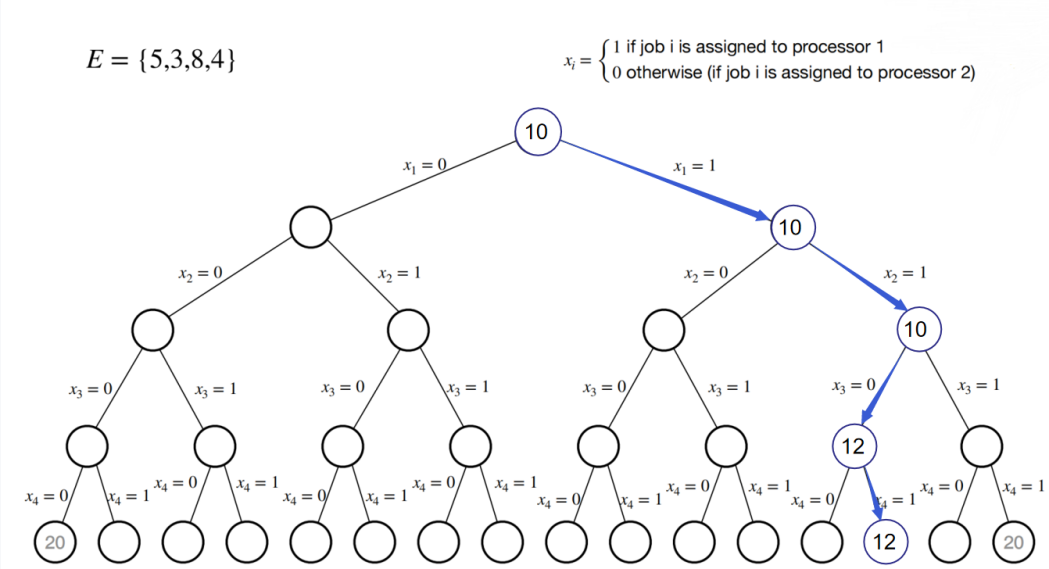
*x*1 = 1, *x*2 = 1, *x*3 = 0时 the remaining jobs is{4},,

,

;

*x*1 = 1, *x*2 = 1, *x*3 = 0, *x*4 = 1时，

, *LB* = 12;

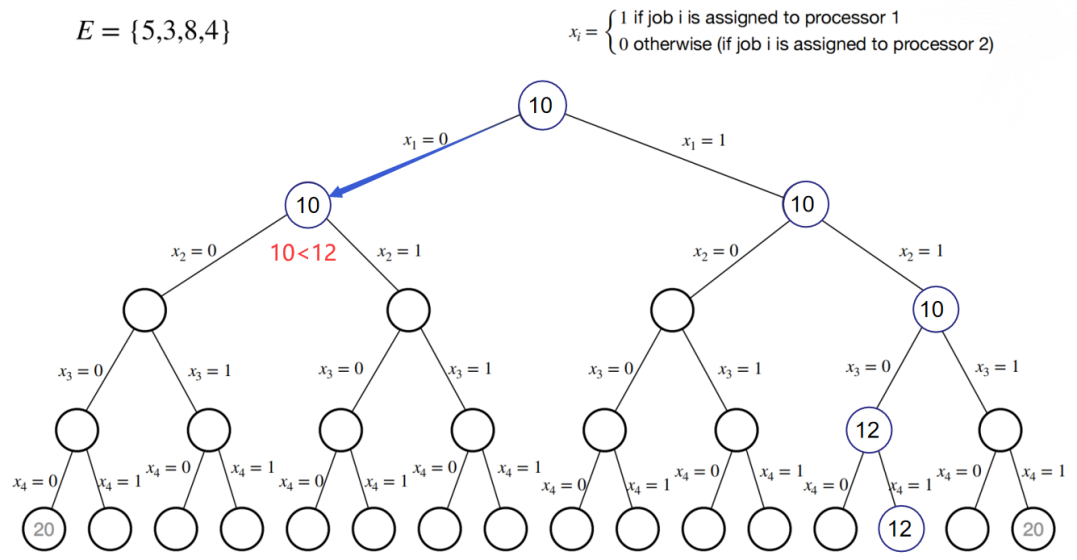


接着，看*x*1 = 0的branch，

*x*1 = 0时 the remaining jobs is{3,8,4},,

5

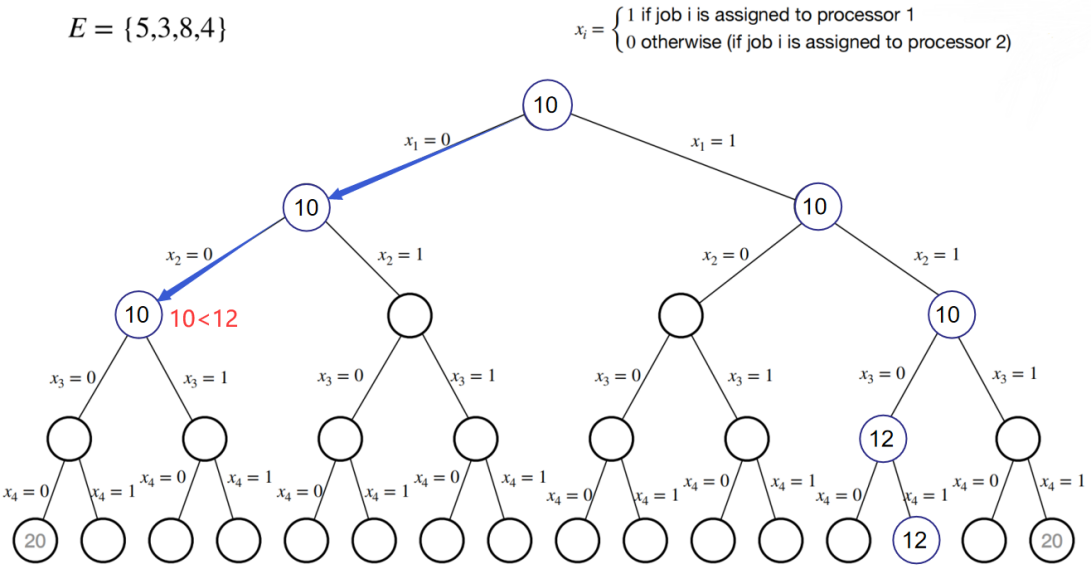
;



10<12继续深度遍历,看*x*1 = 0, *x*2 = 0时,the remaining jobs is{8,4},,

,

;

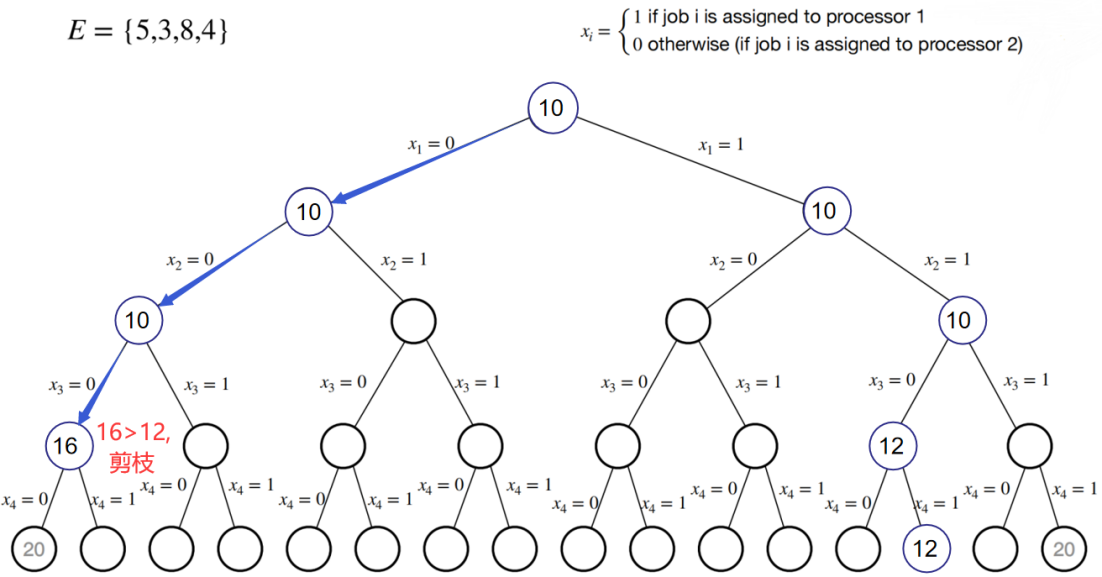


10<12继续深度遍历,

看*x*1 = 0, *x*2 = 0, *x*3 = 0时,the remaining jobs is{4},,

,

;

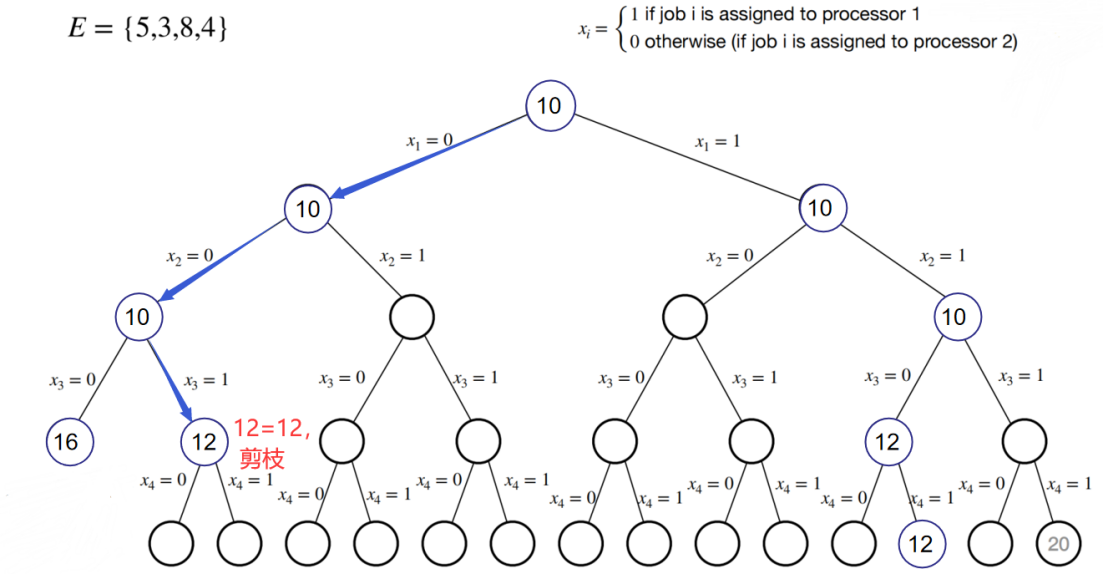


16>12从而进行广度遍历,

看*x*1 = 0, *x*2 = 0, *x*3 = 1时,the remaining jobs is{4},,

,

;

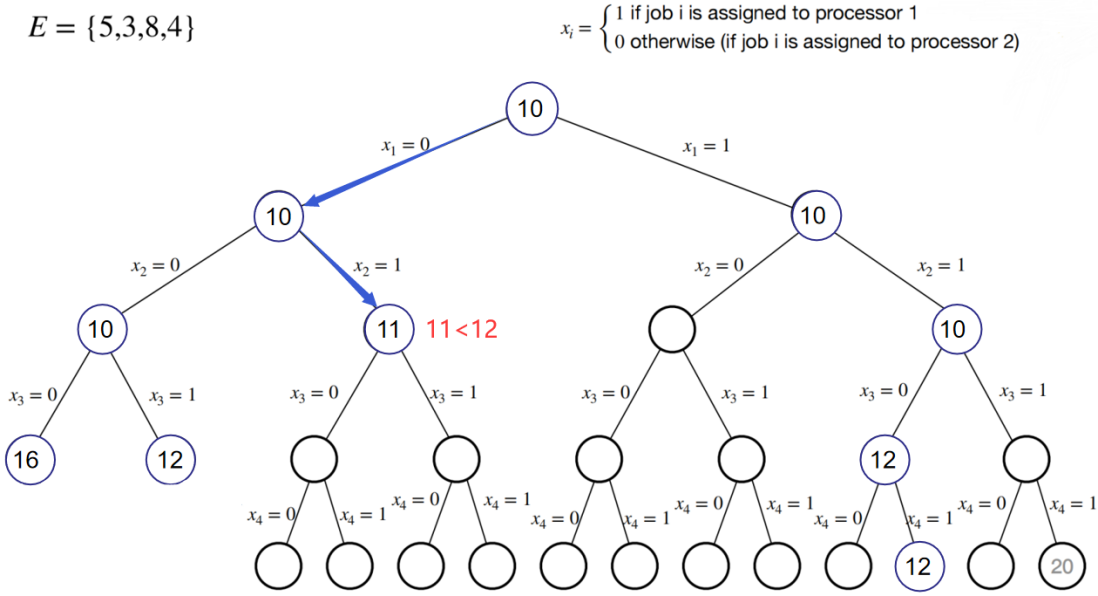


12=12从而进行广度遍历,

看*x*1 = 0, *x*2 = 1时,the remaining jobs is{8,4},,

,

;

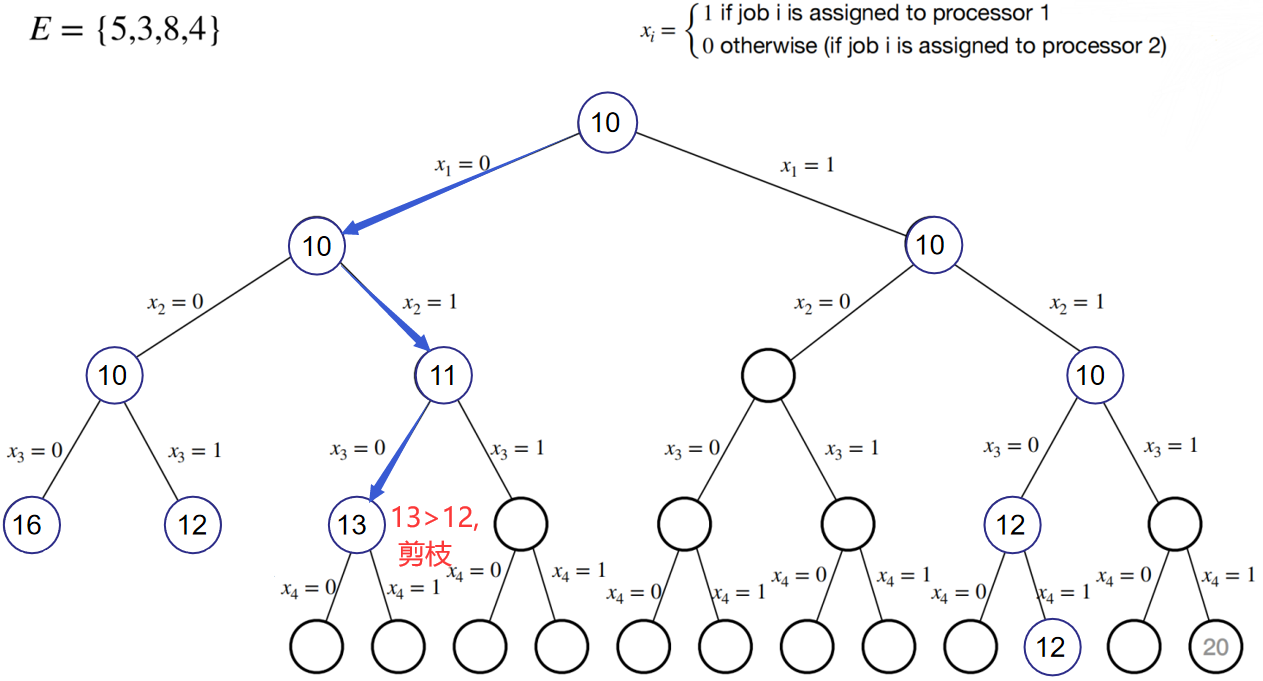


11<12继续深度遍历,

看*x*1 = 0, *x*2 = 1, *x*3 = 0时,the remaining jobs is{4},,

,,

;

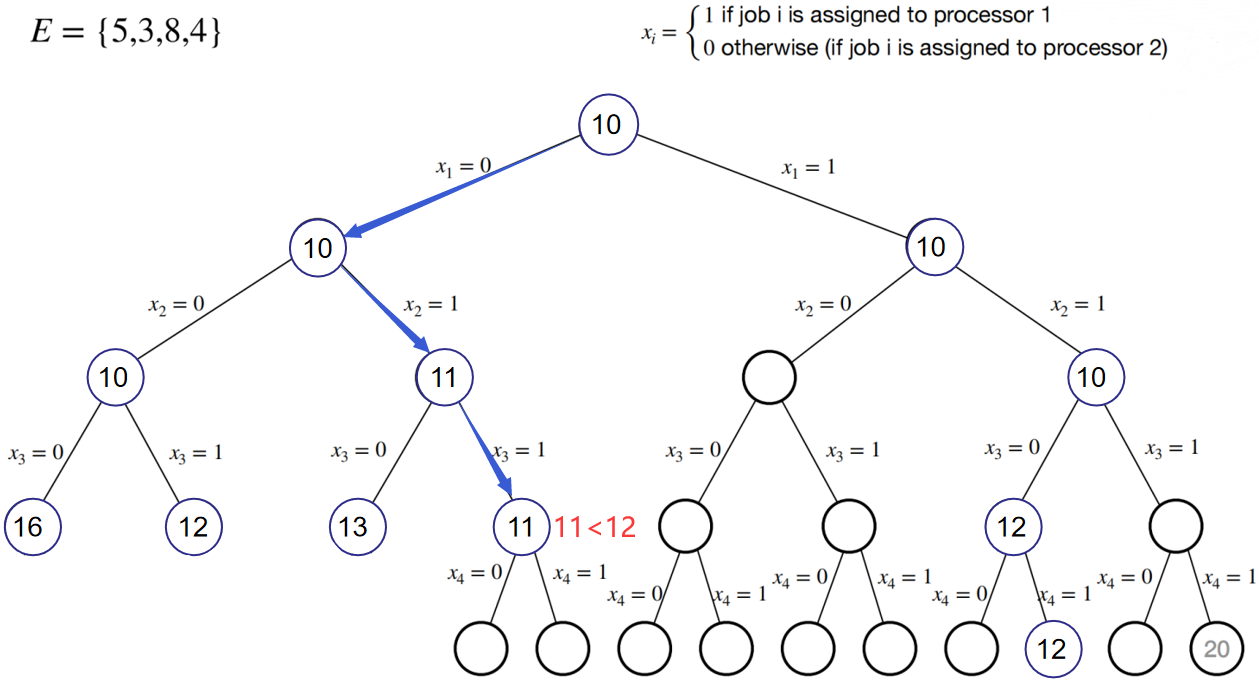


13>12从而进行广度遍历,

看*x*1 = 0, *x*2 = 1, *x*3 = 1时,the remaining jobs is{4},,

,,

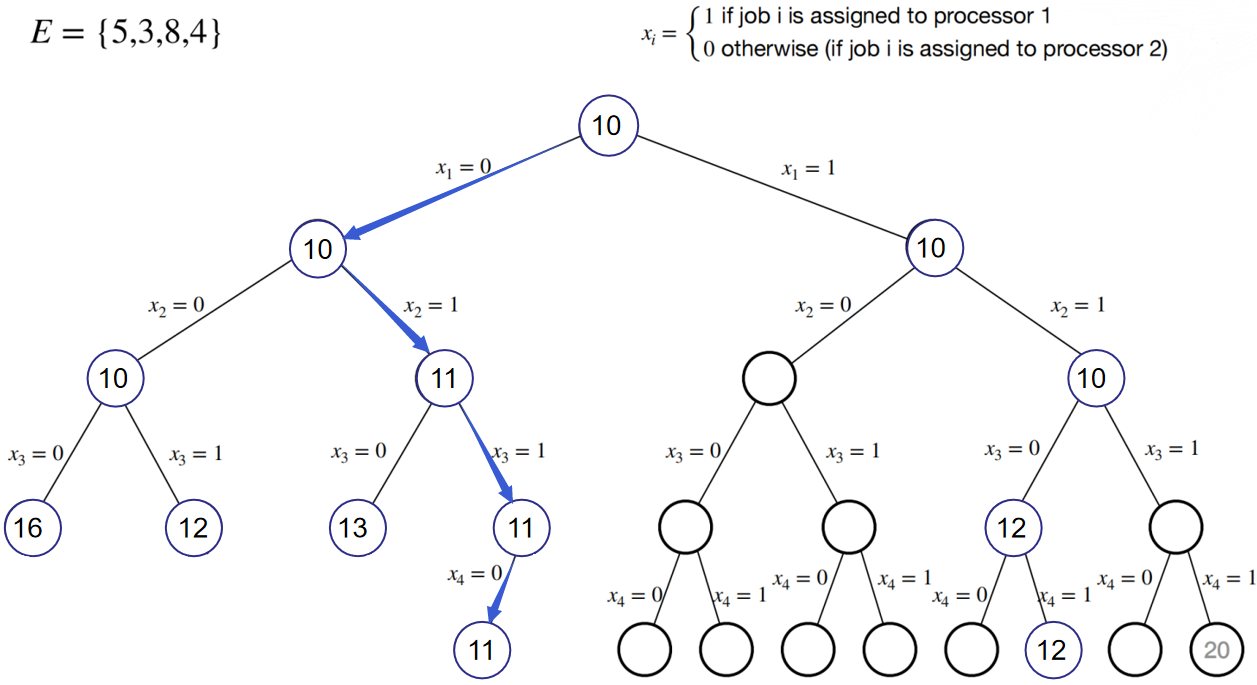
;



11<12继续深度遍历,

看*x*1 = 0, *x*2 = 1, *x*3 = 1, *x*3 = 0时,

, = 11;



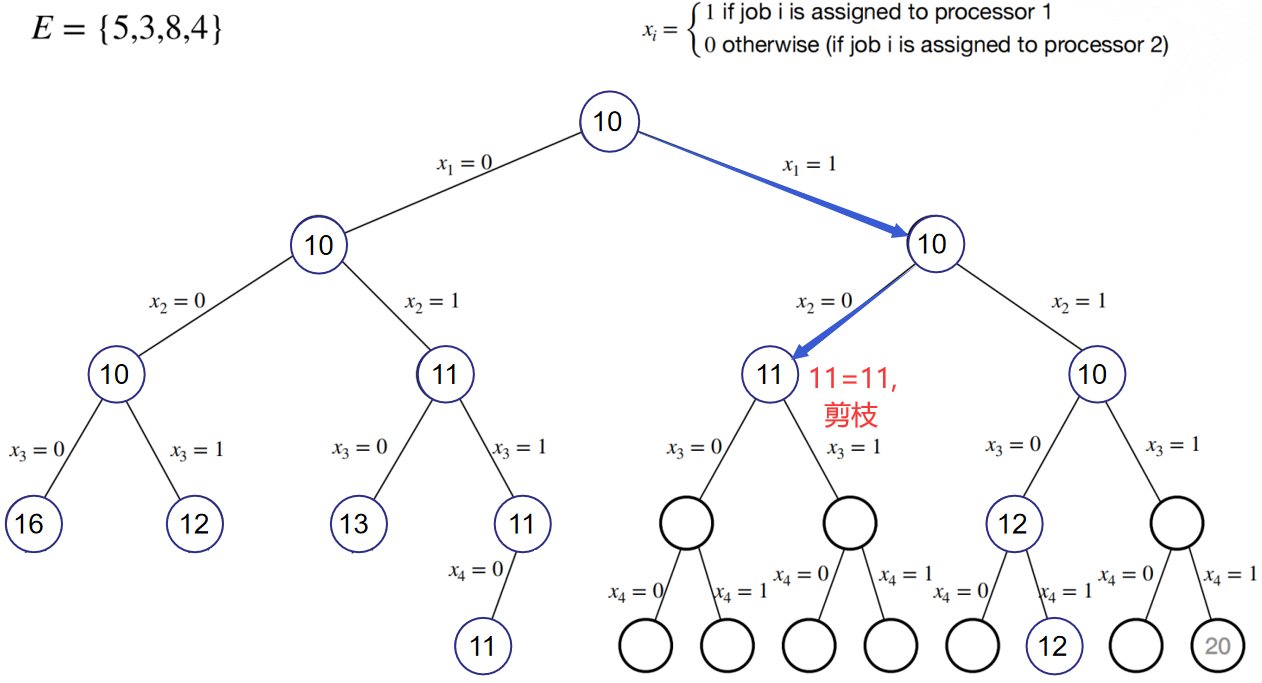
从而得到一个左子树的最优解*x*1 = 0, *x*2 = 1, *x*3 = 1, *x*3 = 0；

下面，看*x*1 = 1的branch，

看*x*1 = 1, *x*2 = 0时,the remaining jobs is{8,4},,

,

;

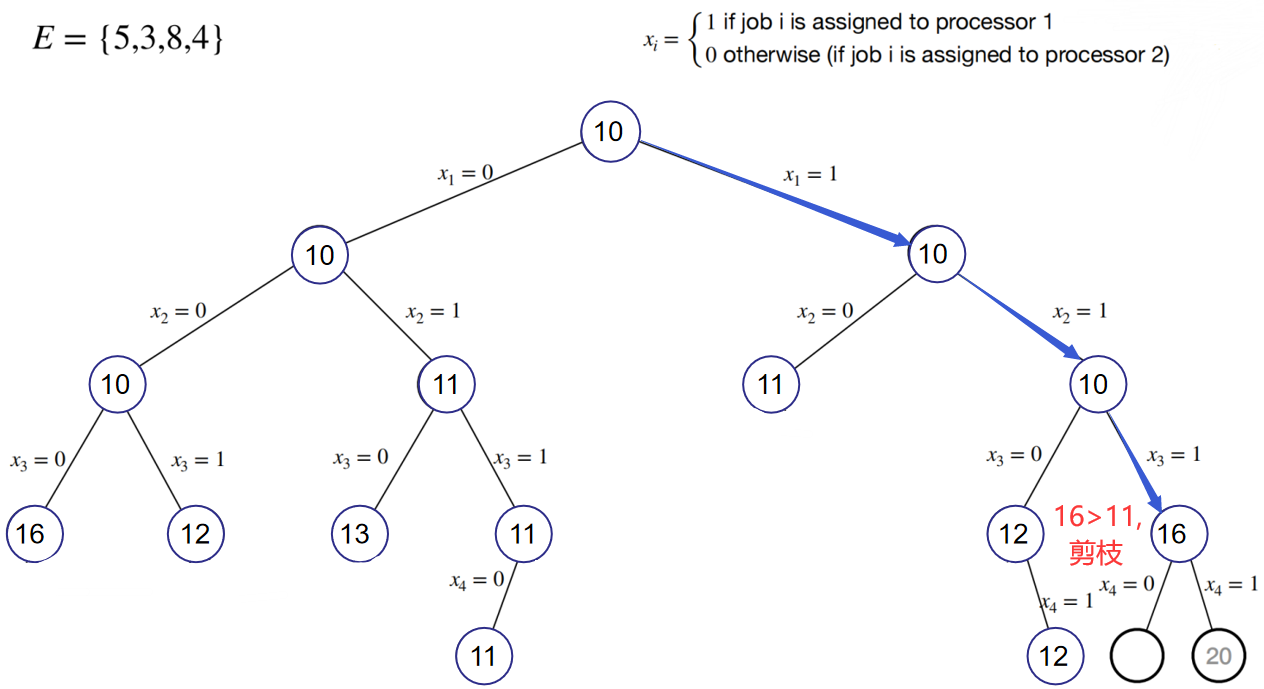


11=11从而进行广度遍历,

看*x*1 = 1, *x*2 = 1, *x*3 = 1时,the remaining jobs is{4},,

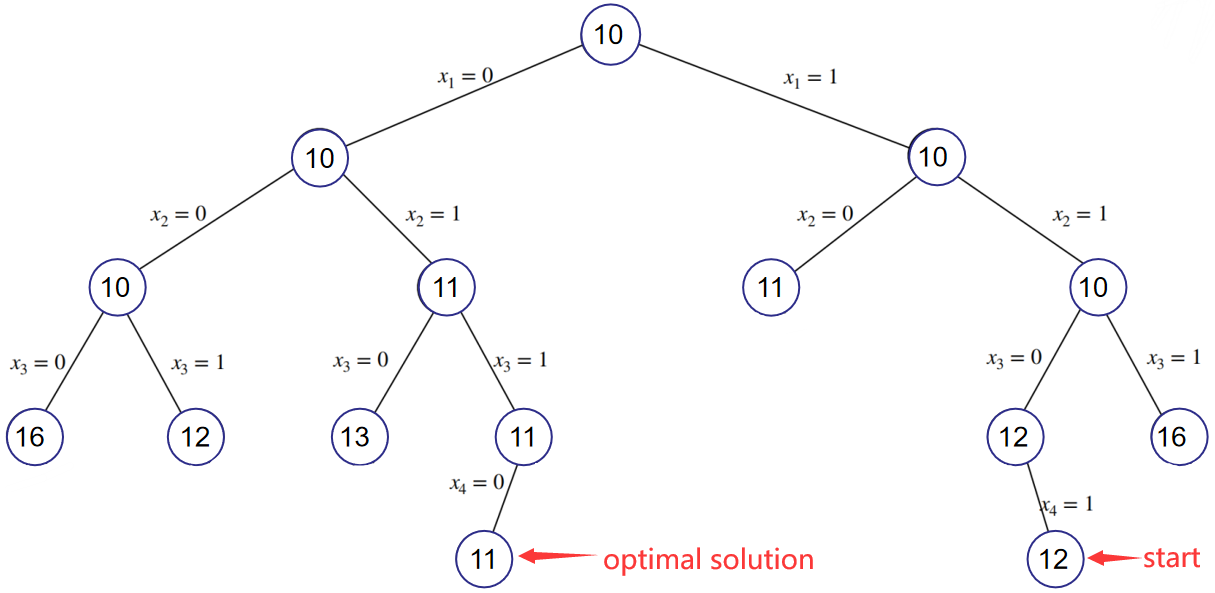
,

;



从而得到最优解为*x*1 = 0, *x*2 = 1, *x*3 = 1, *x*4 = 0;

The tree of the Branch & Bound:



The optimal solution:

processor 1 : 3 8

processor 2 : 5 4

即 minimum load=11;

1. The order of the nodes I explored:

*x*1 = 0, *x*2 = 1, *x*3 = 1, *x*4 = 0