§7.5 回溯算法

\*有时我们处理找最优解的问题，没有任何理论帮我们找最优解，我们只能求助于穷举搜索方法。这一节我们介绍一种系统化的穷举搜索技术，称为回溯技术，并介绍一种称为剪枝的技术，它经常能帮我们减少搜索的工作。

一．例子：

以象棋、围棋或三子棋为例。

有两个对弈者，轮流下子(动作)，比赛的状态由棋盘上棋的位置决定。棋有某些规则确定它何时终止。我们可以把比赛与一棵树对应，该树称为博弈树。

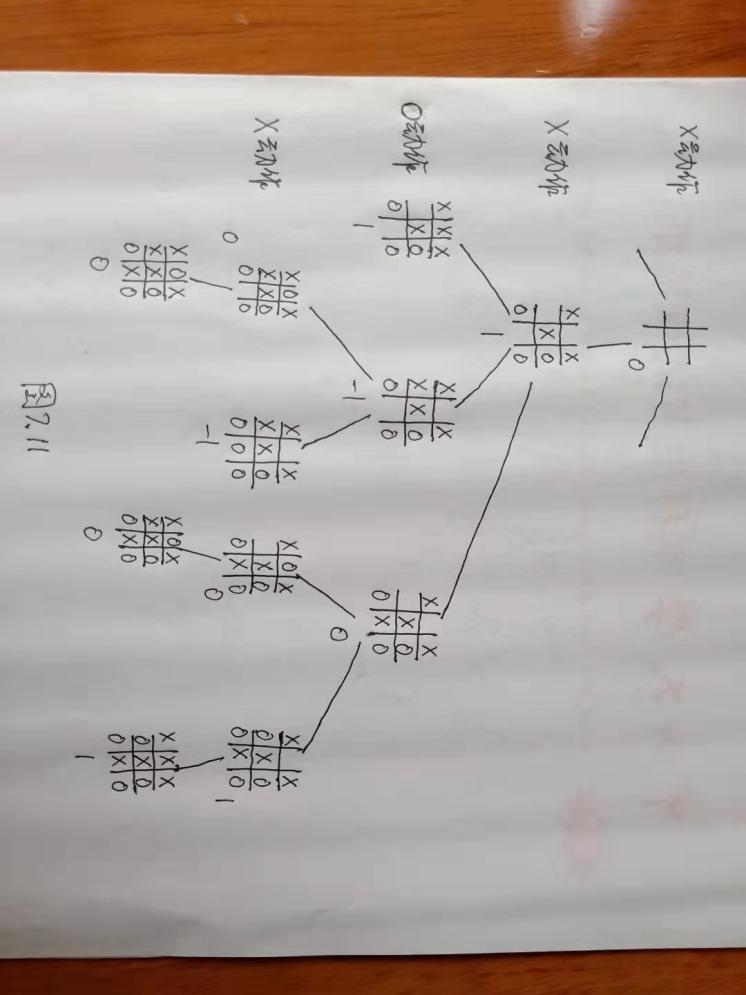
以三子棋为例，博弈树见图7.11。

图7.11

规则：

1.首先给每个叶子赋一个数，如果X博弈者赢，则赋1；如果平，则赋0；如果X博弈者输，则赋。

2.如果结点n的每个孩子已赋了数，则给结点n赋数。如果结点n轮到X博弈者动作，则取n的所有孩子赋的数的最大者赋给n结点;如果结点n轮到O博弈者动作，则取n的所有孩子赋的数的最小者赋给n结点。

3.如此下去，直到每个结点都有值。

4.如果某结点对应X博弈者动作，则他按值最大的孩子动作；如果某结点轮到O博弈者动作，他按值最小的孩子动作。

例如：图7.11，当某结点轮到X动作，且该结点的值为1，则X有赢的策略；当某结点轮到O动作，且该结点的值为，则O有赢的策略。

二．报偿函数

1.报偿函数：上述结点的值0, , 1，可以推广为赋给结点任一数，称为报偿函数。

规则同上：如果某结点轮到参加者1动作，该结点的值取其所有孩子的值的最大者；如果该结点对应参加者2动作，则该结点的值取其所有孩子的值的最小者。

2.报偿函数的应用：用来估计下一步动作输、赢的可能性。

例如：围棋、象棋状态太多，计算机内存无法容纳整棵博弈树，可向前看几步，看看输赢的可能性。

三．实现回溯查找

FUNCTION Search (B : boardtype; mode : modetype) : real;

/\* 计算B状态下的报偿函数值。如果 mode=MAX, 则轮到参加者1动作；如果 mode=MIN，则轮到参加者2动作；返回报偿函数值 \*/

VAR

C : boardtype;

value : real;报偿值

BEGIN

1. IF B is a leaf THEN
2. RETURN( payoff(B) )

ELSE BEGIN

1. IF mode = MAX THEN
2. value :=
3. ELSE value := ;
4. FOR each child C of board B DO
5. IF mode = MAX THEN
6. value := max( value, Search(C, MIN) )

ELSE

1. value := min( value, Search(C, MAX) );
2. RETURN(value)

END

END;

四．剪枝

\*有时在回溯查找中，可以不考虑某些结点。

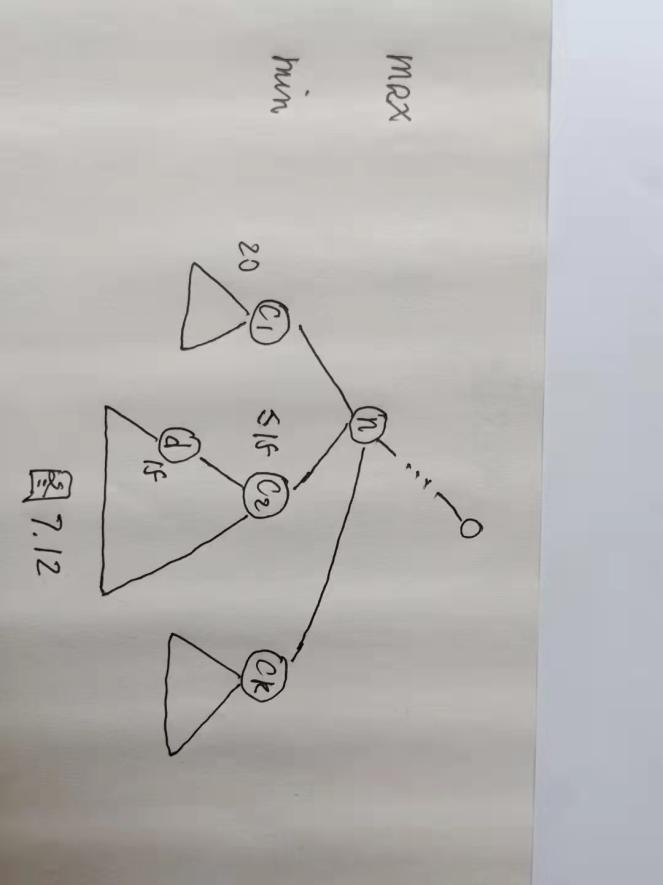
例子：

图7.12

图7.12中的d以外的其它孩子可以不考虑(进行剪枝)。

1.最终值与暂定值

最终值是前面所述的赋给每个结点的值。暂定值是min结点的值的某个上界，或者是max结点的值的某个下界。

2.计算最终值和暂定值的规则：

(1) 如果结点n的所有孩子或者已考虑过，或者已被剪枝，则n的暂定值就是最终值。

(2) 如果max结点n有暂定值, 它的一个孩子有最终值, 那么n的暂定值改为。如果n是min结点，则把它的暂定值改为。

(3) 如果p是min结点，其父结点为q(max结点)，并且p和q的暂定值分别为和，并且，那么我们可以剪除p的所有未考虑的孩子。在且p是max结点的情况下，我们可以剪除p的所有尚未考虑的孩子。

3.例子

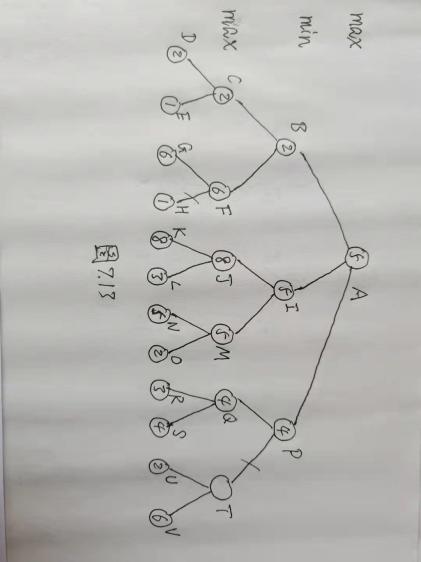


图7.13

五．批处理作业调度

1.问题描述

给定n个作业的集合。每个作业都有两项任务分别在两台机器上完成。每个作业必须先由机器1处理，然后由机器2处理。作业需要机器j处理的时间为，其中，。

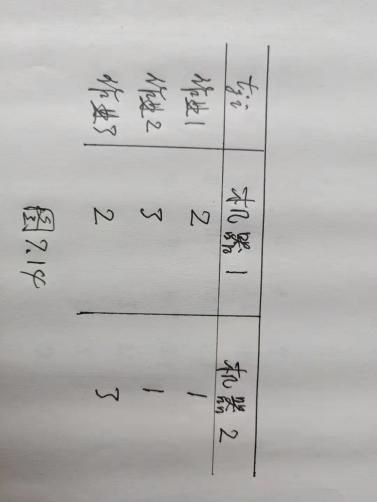
对于一个确定的作业调度，设是作业在机器上完成处理的时间。所有作业在机器2上完成处理的时间和称为该作业调度的完成时间和。

批处理作业调度问题要求对于给定的n个作业，制定最佳调度方案，使其完成时间和达到最小。

\*批处理作业调度问题的一个常见例子是在计算机系统中完成一批n个作业，每个作业都先完成计算，然后将计算结果打印输出。计算任务由计算机的中央处理器完成，打印输出任务由打印机完成。在这种情况下，计算机的中央处理器是机器1，打印机是机器2。

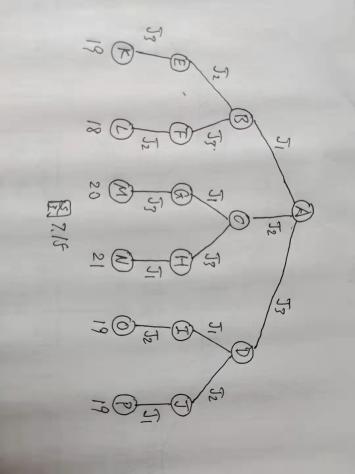
对于批处理作业调度问题，可以证明，存在最佳作业调度使得在机器1和机器2上，作业以相同次序完成。因此，我们求这样的作业调度。

2. 例子：的实例。

图7.14

3个作业的6种可能的方案：1, 2, 3; 1, 3, 2; 2, 1, 3; 2, 3, 1; 3, 1, 2; 3, 2, 1。它们的完成时间分别是：19, 18, 20, 21, 19, 19。

回溯法搜索排列树为：

图7.15

用回溯法求最优安排，要遍历整棵树，找完成时间的最小叶子。共需时间。

作业16：

1.考虑一个游戏，有六个石子，参加者1和2轮流拿石子，每次拿1至3个石子。最后拿石子的那个人输。

a) 画出这个游戏的整棵博弈树；

b) 如果该博弈树用剪枝技术查找，结点代表棋局，先查找取最少石子的分枝，那么哪些树枝要剪除？

c) 如果每个参加者都采取最佳策略，谁赢得这场游戏？