

Pair Project (2048) Report

Xizhou (v-xizzhu)

2014/9/30

这次 2048 结对编程的项目我是独立完成的，考虑到所需实现的功能较简单，没有写单元测试。

花了大概 3 天时间调研，起初考虑是 alpha-beta 剪枝，但事实上我的对立面是随机的而不是有智能的，构不成博弈，alpha-beta 会太过保守，只考虑几步内使得最糟的不至于太糟，就可能会错失长远的利益。

我最终的方法还是搜索，不过是我方选局面评分最好的移动方向，对方根据 2、4 随机出现的概率算局面的期望评分，这样基本上其实就是穷举法了，所以适当的根据概率作剪枝，只把一些小概率出现的局面给剪去了（比如一直出现 4）。

对局面的评分，参考了网上大家提出的各种 2048 的策略，主要就是有序，可以是直角排序可以是线型排序或者是蛇形排序，事实上对于任何行、列，保证某一个方向有序就可以。为了希望结果更好，也就是希望对比搜索深度更深的局面也有一个好的保障，将有序性改为要求严格有序（如果逆序便减分），相等便加分（可以在之后的移动里将两个数合并），此外还加入了空闲格越多分越高（适当的改善评分，权重不能太大，不然就会由于搜索深度不够深，过早地无脑合并数字了），最后还加入了一项评分数字越大得分越低，并且两数合并之后的评分高于两数合并之前的评分，数字越大，合并后分差越大，由于局面数字只会越来越大，由此便可以控制其合并尽可能大的数字。假设搜索到某一个局面，各个方向都无法移动（就是 Game Over），那就给这个局面一个比较大的惩罚分数。

为了加快搜索的速度（几乎是穷举会导致局面数量指数增加，对全局面计算要求非常高），我尽可能的使用查表的方式，但是全局面的可能性太多存不下，故此只存一行或者一行可能的情况（限定最大数为 $2^{15} = 32768$ ，理论上假设最后出来的数字是 2，最大数为 $2^{16} = 65536$ ），这样就只需存下 65536 (16^4) 种可能。对行或者列上下左右移动的结果存下来，并且对局面的评分也改为了对每行每列评分之和。为了进一步加速使用了位运算，一个局面只需一个 Int64 就可以存下。

程序的编写大概用了两个晚上，之后一直在调参数和评分形式，考虑过通过机器学习评分，但没头绪放弃了。搜索深度，理论上应该是与最大的需要被合并的数字合并所需步数成正比，实际操作中和局面中不同数值的个数线性相关。使用了置换表、多线程，对初始待搜索局面的四个移动方向并行搜索。

目前的程序，没写命令行版的 2048 测试，由于时间限制，测试的局数不多（大概 10 盘以内），每一局都能玩到 4096，其中只有一局没到 8192，最高一次得分 374296（一个 16384，一个 8192），在公司电脑上跑得蛮快的，但是我忘了数每秒多少步了。