

刘丰瑞, 田少杰, 任玉昕. 动态规划求解状态总数算法在揭棋上的应用[J]. 智能计算机与应用, 2025, 15(6): 151–155. DOI: 10.20169/j.issn.2095-2163.250623

动态规划求解状态总数算法在揭棋上的应用

刘丰瑞, 田少杰, 任玉昕

(北京信息科技大学 计算机学院, 北京 102206)

摘要: 揭棋是中国象棋的一个变种玩法。相较于中国象棋, 揭棋的随机性更强, 棋子移动范围更广, 具有一定研究价值, 其解法仍有待研究。状态总数是衡量棋类问题复杂度的重要指标, 计算出揭棋的状态总数对揭棋的解法研究有重大意义, 对揭棋状态问题的分析也有利于对揭棋的认识和了解。利用动态规划算法可以较为精确地计算出揭棋的状态总数上界, 该值为 1.14×10^{53} 。

关键词: 动态规划; 中国象棋; 揭棋; 组合计数; 状态复杂度; 状态总数

中图分类号: TP301.6

文献标志码: A

文章编号: 2095-2163(2025)06-0151-05

Dynamic programming for solving state totals algorithm on revealed chess

LIU Fengrui, TIAN Shaojie, REN Yuxin

(College of Computer Science, Beijing Information Science and Technology University, Beijing 102206, China)

Abstract: Revealed chess (Jieqi) is a variant of Chinese chess. Compared with Chinese chess, revealed chess is more randomized and has a wider range of piece movement, which has certain research value, and its solution method still needs to be studied. The total number of states is an important index to measure the complexity of chess problems. Calculating the total number of states of revealed chess is of great significance to the study of solution of revealed chess, and the analysis of state problem of revealed chess is also conducive to the knowledge and understanding of revealed chess. The dynamic programming algorithm can be used to calculate the upper bound of the total number of states of revealed chess more accurately, and the value is 1.14×10^{53} .

Key words: dynamic programming; Chinese chess; revealed chess; combinatorial counting; state complexity; total number of states

0 引言

揭棋是中国象棋的一个变种玩法^[1-3], 相较于中国象棋而言, 揭棋的变化更多, 棋子活动范围更大, 能够产生的局面也更多^[4-5], 这种玩法有更高的趣味性和随机性。

揭棋的状态总数是指在规则允许的范围内棋盘上能产生的局面总数, 这个概念也被一些学者称为状态复杂度^[6-8]。

棋类游戏的状态总数和博弈树总数是衡量棋类复杂度的重要指标, 棋类复杂度又决定了针对这种棋类所采用的算法。研究一些求解棋类状态总数的算法, 有利于判断棋类问题的复杂度, 也有利于对棋

类问题的求解^[8]。

魏印福等学者^[7]利用动态规划求解了中国象棋的状态总数, 那么将这种求解思路也移置中国象棋的变种玩法、揭棋上是否适用无疑是值得讨论和研究的。

1 问题描述

1.1 棋子摆布与胜负判定

中国象棋棋盘上有 9 条竖线、10 条横线, 总共有 90 个位置^[9-10]。棋子分为红色与黑色两种, 每种颜色 16 枚棋子, 包括“车”、“马”、“炮”、“相”、“士”、“将”、“兵”这 7 种, 而揭棋的棋盘与棋子都与象棋相同。

基金项目: 北京信息科技大学促进高校分类发展-大学生创新创业训练计划项目(5112310855)。

作者简介: 刘丰瑞(2003—), 男, 本科生, 主要研究方向: 棋类机器博弈, 状态复杂度研究。Email: 2021011194@bistu.edu.cn; 田少杰(2002—),

男, 本科生, 主要研究方向: 前端设计; 任玉昕(2001—), 男, 本科生, 主要研究方向: 软件工程。

收稿日期: 2023-11-10

哈尔滨工业大学主办 ◆ 专题设计与应用

揭棋遵循中国象棋的胜负判定规则^[11]。同时,揭棋的棋子摆布也与中国象棋的摆法相同,只不过除“将”、“帅”外的棋子均需要翻面并随机摆放。开局时的摆放形式如图 1 所示。

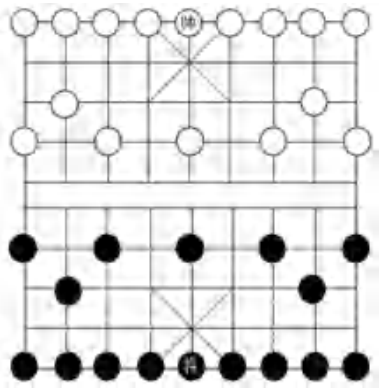


图 1 揭棋开局的棋子摆布
Fig. 1 The start state of revealed chess

1.2 棋子走法

暗子的走法与其位置有关,该子在中国象棋中哪个棋子的初始位置上,那么暗子就要按照这个棋子的走法来行棋,暗子走过一步后需要翻面变成明子。暗子移动后的棋局如图 2 所示,在图 2 中,红棋“兵”位置上的暗子 1 按照“兵”的走法向前进一步,翻开后明子为“车”,威胁到了黑棋“卒”位置上的暗子 3,于是黑棋以“马”的走法移动暗子 2,翻开后明子为“卒”,恰好可以保护暗子 3。

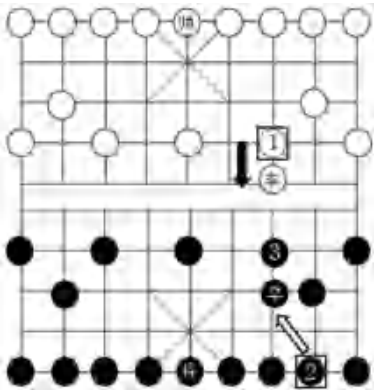


图 2 暗子移动后的棋局示例
Fig. 2 Example of a game after moving a dark disc

揭棋中,明子的走法与中国象棋中的棋子走法基本一致。除了“士”、“象”可以过河外,走法与中国象棋中同名棋子走法完全一致^[12]。

1.3 棋子的选位范围

1.3.1 “将”、“帅”的选位范围

对于红棋和黑棋双方来说,“将”、“帅”的可选位置是关于河界镜像对称的,所以在这里只选取“帅”的摆放方式来进行讨论。由于“帅”只能在九

宫格内活动,因此“帅”只有 9 个位置可走^[7],具体的位置及其编号如图 3 所示。“将”、“帅”的选位无疑是最重要、也最应该优先考虑的,没有“将”或者没有“帅”的棋盘就绝对不能算作有效的局面。

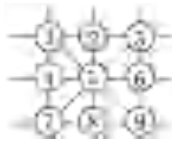


图 3 “帅”的可走位置与对应编号
Fig. 3 Walkable position with corresponding number of the Shuai

1.3.2 暗子选位范围

暗子的选位与“将”、“帅”选位一样关于河界对称,所以在这里只选取红棋部分的暗子摆放方式来讨论。暗子离开初始位置后就会翻面变成明子,因此暗子的可选位置实际上只有初始的 15 个位置,一旦离开这些位置,暗子状态就会被永久性改变。暗子的可选位置与相应编号如图 4 所示。图 4 中的 11 号、12 号位置与图 3 中的 7 号、9 号位置重合。因此,当“将”、“帅”在 7 号或 9 号位置时,暗子的可选位置只有 14 个。

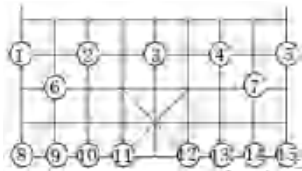


图 4 暗子的可走位置与对应编号
Fig. 4 Walkable position with corresponding number of dark discs

1.3.3 “兵”、“卒”的选位范围

为了便于表述,接下来只讨论“兵”的情况,“卒”的情况与“兵”同理。在揭棋中,“兵”的选位范围实际上是整个棋盘。过河后的“兵”可以平移与向前,想要到达另一半棋盘中的任何一个位置都不困难。在这里主要讨论的是“兵”未过河、只能向前的情况,因此只讨论己方的半个棋盘便足够了。利用“车”位置上的暗子就可以让“兵”到达己方棋盘的任意位置了,具体情况如图 5 所示,在底线无子的情况下,“车”位置的暗子可以轻易地平移到编号 1~8 的 8 个位置,假设这 8 次平移每次翻开后出现的明子都是“兵”,那么“兵”只要继续向前拱到河界就遍历了这一路的全部位置。至于“暗车”那一路,只需要将暗子放到位置 8、此时该暗子恰好处于另一个“车”位,再清空底线,把位置 8 的暗子平移到该路,再恰好翻出一个“兵”,那么这一路也会被“兵”遍历。综上所述,既然己方棋盘的每一路都有可能被“兵”遍历,那就代表“兵”的理论位置遍布整个棋盘。

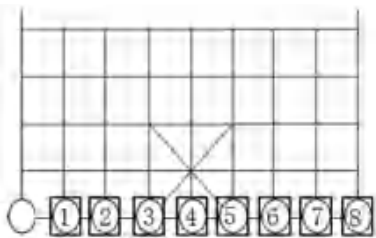


图 5 “兵”遍历己方棋盘的方法

Fig. 5 The way of traverse our board by the Bing

但这是否代表可以随意将“兵”摆放到棋盘上呢? 至少己方底线上同时有 5 个己方的“兵”就不行; 只有“车”位置、“炮”位置上的 4 个暗子可以走到己方底线上的位置, 哪怕这 4 个暗子都翻出“兵”也不能满足要求。由于“兵”向前拱就无法回头, 因此己方底线上永远也不会同时出现 5 个“兵”。

“兵”向前拱之后的状态某种程度上与暗子翻开后的状态一样, 都是永久性的改变, 因此要优先考虑“兵”的摆放。

1.3.4 “象”的选位范围

理论上讲, “象”的可选位置也是整个棋盘。但“象”很容易被“塞象眼”导致无法到达目标位置, 由于暗子的状态改变是永久性的, 所以无法通过移开暗子再让暗子回位的方式让开“象眼”或“象位”。

同时, 因为“象”有 8 组“象位”, 需要考虑已经移动的暗子能够到达的“象位”有哪几组, 这样才能确定“象”的选位范围。“象”的 8 组选位如图 6 所示。



图 6 “象”的 8 组选位

Fig. 6 Eight sets of selections for the Xiang

1.3.5 “士”的选位范围

在没有阻挡的情况下, 1 个“士”可以遍历棋盘上的 45 个位置, 但“士”也会出现被暗子阻挡的情况。

假如开局时, 己方走了一步“士”位置上的暗子, 将其翻开后恰好是个“士”, 但由于己方“兵”位

置上的暗子的阻碍, 尽管这个位置在其活动范围中, 这个“士”也无法到达 1 号位置, 该局面如图 7 所示。

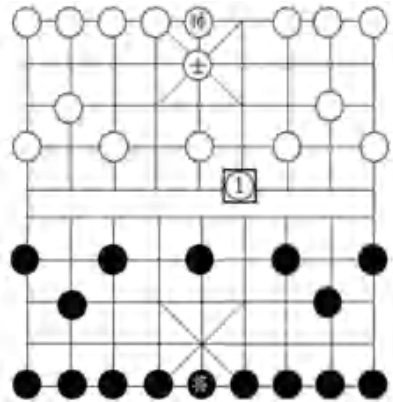


图 7 “士”被暗子阻挡的示例

Fig. 7 Example of the Shi blocked by dark discs

1.3.6 “车”、“马”、“炮”的选位范围

“车”、“马”、“炮”在中国象棋中可以任意行棋, 但在揭棋中仍旧会有特例, 导致“车”、“马”、“炮”被困在某个范围内, 该情形如图 8 所示。

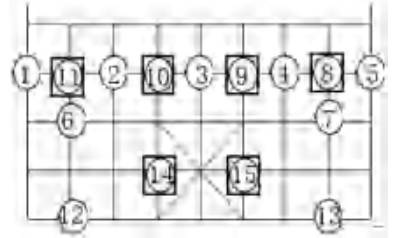


图 8 “车”、“马”、“炮”被阻挡示例

Fig. 8 Example of Ju, Ma and Pao blocked

在图 8 中, 圆形的是暗子, 有矩形框的是被标明的位置。其中, 暗子 7 处于“炮”位, 可以向前一步, 占据位置 8, 假设将其翻开后的明子是“兵”。同理, 暗子 6 可以占据位置 11, 翻开之后也可能是“兵”。暗子 12 处于“马”的位置, 可以走到位置 14。同理, 暗子 13 也可以走到位置 15。假设这 2 个暗子翻开后都是“兵”, 并且一路向上行棋到了位置 9 和位置 10。那么, 此时位置 1~5 上都是暗子, 位置 8~11 上都是“兵”, 而“车”、“马”、“炮”都无法在不吃子的情况下越过这排棋子, 这排棋子也不能移开, 因为一旦移开就不能再回到现在的状态。因此, 在这个状态下, “车”、“马”、“炮”的选位范围只能是这排棋子上方或下方的全部空位, 无法走遍整个棋盘。

2 求解揭棋状态总数上界

2.1 动态规划算法在揭棋中的适用性分析

动态规划求解象棋状态总数的原理, 是把数量

巨大的不同摆放形式的棋盘进行归一化,再通过先难后易的顺序构建树结构进行计算。由于进行了归一化,树的节点数大大减小,时间开销也随之大幅减少。

如上文 1.3 节所述,揭棋中有很多特殊局面需要考虑,每一种特殊局面就代表一个额外的结点,代表更多的时间开销,而每增加一种特殊情况,节点的数量都会指数级增加,这是不可接受的,想要准确排除掉不合理的特殊局面也是耗费时力的。因此,想要直接将该算法应用于准确求解揭棋的状态总数几乎是不可能的。

为了让计算机能够计算出一个有效的结果,可以适当忽略一些不合规则的特殊局面,从而计算出一个相对接近正确结果的上界。下文将忽略“车”、“马”、“炮”、“象”、“士”、“兵”选位受限制的情况,基于明子可以遍布整个棋盘的前提下,对揭棋状态总数上界进行计算。

2.2 上界求解思路

当一个问题由于规模较大而无法解决时,可以将其拆分成一个一个同类型的小问题,按照由特殊到一般的顺序进行处理,这就是动态规划求解状态总数的大致思路^[10]。

在揭棋中,“将”和“帅”是最受限制的棋子,只能在己方的九宫格内活动,暗子则只有 15 个初始位置可选,一旦离开初始位置就不能回到暗子状态。所有其他棋子理论上都可以走到棋盘上任意的位。哪怕是在本方棋盘上机动性极差的“卒”,也可以凭借“车”位置暗子的机动性出现在任意位置。所以,只要摆放好“将”、“帅”和暗子,其他棋子的状态总数基本上就只与棋盘上的空位置数有关了。

综上所述,只需要按照正常计数问题的思路,先做分步乘法,再做分类加法就可以求出揭棋状态总数的上界了。

2.3 具体实现

在这个实现的过程中,要保证“将”、“帅”一定是在棋盘上的,否则棋局是无效的,但考虑到实现的复杂性以及效率问题,棋局中有“‘将’、‘帅’对脸”的结果并未被剔除。

同时,有些局面是在正常对局中不可能出现,但实际存在的,而这也会被算作合理局面。

为了便于表示,本文中会定义一些变量,为了节约篇幅,变量下角标为 R 代表红方,下角标为 B 代表黑方。

2.3.1 “将”、“帅”摆放

“将”只有 9 种位置,在这 9 种位置中有 2 种会影响暗子的选位,那就把“将”的选位分为 2 类。其中一类会影响暗子摆放,另一类不会影响。由于需要同时考虑红黑两色的“将”、“帅”,最终可以由排列组合得知有 4 种可能的情况。为了方便表示,可以定义一些变量,具体定义见表 1。表 1 中, K_R 与 K_B 值为 0 或 1, 2 个不同的值代表 2 种不同的状态,即“将”、“帅”是否影响暗子摆放。

表 1 部分变量及其释义

Table 1 Selected variables and their meanings

变量名	变量含义
K	“将”、“帅”当前位置所属的状态种类
Tk	在 K 状态下“将”、“帅”的可选位置总数
Ta	所有暗子与明子在 K_R 和 K_B 状态下的摆放方式数量
S	揭棋的状态总数

状态总数的计算公式为:

$$S = \sum_{K_R=0}^1 \sum_{K_B=0}^1 T_{k_R} \cdot T_{k_B} \cdot Ta \tag{1}$$

其中的变量大多数都是已知的,只有 Ta 的值需要通过暗子与明子的摆放来确定。

2.3.2 暗子摆放

每一方都有 15 个暗子,每一方的暗子最多都只有 15 个位置可选,如果“将”、“帅”占了一个暗子的位置,那么暗子就只有 14 个位置可选,具体的位置数量要根据 K_R 和 K_B 的值来确定。

为了便于说明,继续定义一些变量见表 2。

表 2 部分与暗子相关的变量及其释义

Table 2 Some variables about dark discs and their interpretation

变量名	变量含义
A	暗子的数量
Pd	暗子的可选位置数量
Tr	其他明子在“将”、“帅”、暗子摆放好后共有几种摆放方式

研究推出 Ta 的计算公式为:

$$Ta = \sum_{A_R=0}^{Pd_R} \sum_{A_B=0}^{Pd_B} C_{Pd_R}^{A_R} \cdot C_{Pd_B}^{A_B} \cdot Tr \tag{2}$$

其中的 C 是一个数学符号,意为从 U_1 个元素中取出 U_2 个数字,总共会有 $C_{U_2}^{U_1}$ 种取法。

2.3.3 明子摆放

由于不同种类的明子在相同位置会产生不同的棋局,因此定义一些表示明子数量的变量,考虑到所有明子理论上的选位范围都是棋盘上的任意空位,

定义一个代表空余位置的变量名也是必要的,具体的变量定义见表 3。同时,由于明子是由暗子翻开得来,所以任意一方的暗子与明子的总数都不应超过 15 个。

表 3 变量与函数的释义

Table 3 Interpretation of variables and functions	
变量或函数名	变量或函数含义
M	明子的数量
$Blank$	棋盘上空位的数量
$getMing$	在红、黑棋的“将”、“帅”与暗子固定后,求在有 M_R 个红色明子、 M_B 个黑色明子、 $Blank$ 个空位的情况下这些棋子有几种摆放方式的辅助函数

最后推得, Tr 的公式定义如下:

$$Tr = \sum_{M_R=0}^{15-A_R} \sum_{M_B=0}^{15-A_B} getMing(Blank, M_R, M_B) \quad (3)$$

其中通过棋盘上的总位置数减去“将”、“帅”与暗子数量之和,同时再减去前面所有乘积项的上标之和即可得到 P 的值。此时,已经求得了式(1)及其衍生式中的所有位置数,至此也就求出了 S 值。

2.3.4 求解结果

根据上述思路所编写的程序得出了揭棋的状态总数约为 1.14×10^{53} , 相比中国象棋的 $7.54 \times 10^{39.88}$ 大很多。

3 结束语

本文利用动态规划算法计算出了揭棋状态总数上界为 1.14×10^{53} , 并得到了该算法不适用于精确求解揭棋状态总数的结论。可以预见的是,该算法适用于棋子可移动、且容易进行局面归一化的棋类游戏,利用该算法只需进行组合计算并累加即可得到

状态总数。

利用动态规划算法计算揭棋状态总数的下界无疑也是可行的。在后续工作中将会剔除部分有效局面,计算出下界具体值,更加精确地得到揭棋状态总数的估计值。

参考文献

[1] FAN Shaoze, ZHANG Shun, LIU Jianbo, et al. Power converter circuit design automation using parallel Monte Carlo tree search [J]. ACM Transactions on Design Automation of Electronic Systems, 2023, 28(2): 17–33.

[2] BORY P. Deep new: The shifting narratives of artificial intelligence from deep blue to AlphaGo [J]. Convergence: The International Journal of Research into New Media Technologies, 2019, 25(4): 627–642.

[3] DONG Peng, LIU Hongcheng, LEI Xing. Monte Carlo tree search based non-coplanar trajectory design for station parameter optimized radiation therapy (SPORT) [J]. Physics in Medicine and Biology, 2018, 63(13): 135014.

[4] 朱舟, 闵华松. 利用统计数据并行蒙特卡罗树搜索算法的中国象棋博弈[J]. 计算机工程与应用, 2024, 60(23): 340–348.

[5] 邱扬廷. 中国象棋游戏中的智能语音识别技术研究及应用[D]. 重庆:重庆理工大学,2024.

[6] 钱希兴. 基于工业机器人的中国象棋对弈系统研究[D]. 大连:大连理工大学,2023.

[7] 魏印福,李舟军. 动态规划求解中国象棋状态总数[J]. 智能系统学报,2019,14(1):108–114.

[8] 高强. 计算机博弈问题的复杂性、理论解及相关搜索算法研究[D]. 沈阳:东北大学,2016.

[9] 马麟. 关于中国象棋的发展论述[J]. 求知导刊, 2016(2):31.

[10] 黄晨. 棋类游戏时的先行权[J]. 智能系统学报, 2007,2(3): 91–94.

[11] 中国人工智能学会机器博弈专业委员会. 亚洲象棋规则 [EB/OL]. (2014–07–16) [2023–10–20]. <http://computergames.caii.cn/jsgz11.html>.

[12] 百度百科. 中国象棋 [EB/OL]. (2023–10–07) [2023–10–20]. <https://baike.baidu.com/item/%E4%B8%AD%E5%9B%BD%E8%B1%A1%E6%A3%8B/278314>.