

# Design and Application of NegaScout Algorithm in Surakarta Game System

Runze Jia Ling Lu\*

University of Science and Technology, Anshan, Liaoning, 114051, China

## Abstract

Computer game is one of the hot research directions in the field of artificial intelligence, which plays an important role in promoting the progress of artificial intelligence. Surakarta, as a kind of board game with strong strategy, has high research value in game research because of its complex situation and changeable moves. In order to improve the search efficiency and decision level of Surakarta game system, the NegaScout algorithm is introduced and optimized. NegaScout is an improved Alpha-Beta pruning algorithm that speeds up the search process by reducing unnecessary branch exploration through a zero-window search mechanism. In this paper, the NegaScout algorithm is implemented in Surakarta game system, and the situation evaluation function is combined to select the best move. The experimental results show that the NegaScout algorithm is superior to the traditional Alpha-Beta pruning algorithm in the search efficiency of Surakarta game, which can guarantee the calculation accuracy and shorten the search time effectively, and provide a more efficient solution for the intelligent confrontation of this game.

## Keywords

Surakarta game; NegaScout algorithm; Alpha-Beta pruning algorithm; game efficiency

# NegaScout 算法在苏拉卡尔塔棋博弈系统中的设计和应用

贾润泽 鹿玲\*

辽宁科技大学, 中国·辽宁鞍山 114051

## 摘要

计算机博弈是当前人工智能领域的热门研究方向之一, 对人工智能的进步具有重要推动作用。苏拉卡尔塔棋 (Surakarta) 作为一种策略性较强的棋类游戏, 因其复杂的局面和多变的走法在博弈研究中具有较高的研究价值。为提升苏拉卡尔塔棋博弈系统的搜索效率和决策水平, 论文引入了NegaScout算法并进行了优化设计。NegaScout是一种改进的Alpha-Beta剪枝算法, 通过零窗口搜索机制减少不必要的分支探索, 从而加速搜索过程。论文在苏拉卡尔塔棋博弈系统中实现了NegaScout算法, 并结合局面评估函数以选择最佳走法。实验结果表明, NegaScout算法在苏拉卡尔塔棋的搜索效率上优于传统的Alpha-Beta剪枝算法, 在保证计算精度的同时有效缩短了搜索时间, 为该棋种的智能对抗提供了更高效的解决方案。

## 关键词

苏拉卡尔塔棋; NegaScout算法; Alpha-Beta剪枝算法; 博弈效率

## 1 引言

在人工智能技术飞速发展的背景下, 机器博弈已成为该领域的重要研究方向, 推动了智能算法在决策和对抗方面的理论创新。机器博弈的技术进展促进了人工智能的应用广度, 当前已在诸如国际计算机奥林匹克、全国大学生计算机

博弈大赛等竞赛中得到了广泛应用, 这些竞赛旨在提升机器博弈算法的实用性和智能水平, 使计算机能够在特定场景下接近甚至超越人类的决策水平。计算机博弈系统通常由棋局表示、搜索算法和局面评估三大核心模块构成, 分别负责描述棋局状态、寻找最佳决策及评估棋局优劣。局面评估往往是设计中的难点, 要求通过量化和模式识别等技术手段来提高计算机的局面判断精度。

目前, 计算机博弈技术在国际象棋、围棋、五子棋等棋类中取得了显著成果, 其中部分棋种的计算机对弈水平已超越了顶尖人类选手。然而, 与这些研究深入的棋类相比, 苏拉卡尔塔棋的计算机博弈研究尚处于起步阶段。苏拉卡尔塔棋因其独特的棋盘布局和复杂规则, 对算法设计提出了更高要求, 尤其在高效搜索算法方面存在挑战。

NegaScout 算法是一种在复杂博弈中应用广泛的优化搜

**【基金项目】** 辽宁科技大学大学生创新创业训练计划项目经费支持 (项目编号: X202410146199)。

**【作者简介】** 贾润泽 (2003-), 男, 中国辽宁庄河人, 在读本科生, 从事计算机应用研究。

**【通讯作者】** 鹿玲, 男, 硕士, 高级工程师, 从事计算机应用、嵌入式开发等研究。

索算法,通过改进 Alpha-Beta 剪枝, NegaScout 可以在减少节点评估的基础上显著提升搜索效率。其核心思想是使用零窗口搜索来优先探索最有可能的主变化路径,从而实现高效剪枝。基于此,论文将研究 NegaScout 算法在苏拉卡尔塔棋博弈系统中的设计与应用,通过构建基于 NegaScout 的博弈模型提升搜索效率,以期为苏拉卡尔塔棋的计算机博弈研究提供高效的技术支持和应用前景。

## 2 苏拉卡尔塔棋概要

苏拉卡尔塔棋 (Surakarta) 是一种源自印度尼西亚的传统棋类游戏,以其独特的棋盘布局和规则在棋类博弈中独树一帜。不同于五子棋、围棋等常见棋类,苏拉卡尔塔棋的棋盘为  $6 \times 6$  的方格,并在棋盘边缘设计了八条环形弧线。这些弧线不仅是棋盘的视觉特征,也是该游戏的关键元素,玩家可以利用弧线进行特殊的绕行移动。这种棋盘设计带来了新颖的战术,极大增加了棋局的复杂性,使苏拉卡尔塔棋在人工智能和计算机博弈研究中具备较高的研究价值。

苏拉卡尔塔棋的对弈双方各有 12 枚棋子,起始位置在棋盘外层。游戏目标是通过吃掉对方所有棋子或使对手无棋可动来获胜。棋子的移动有两种:一种是普通移动,即棋子移动到相邻的空格上;另一种是绕行吃子,玩家可利用棋子绕行弧线并在合适位置吃掉对手棋子。这种绕行规则的加入,使得苏拉卡尔塔棋对棋子的路径规划和吃子时机有较高的策略要求。

苏拉卡尔塔棋的棋局变化复杂度极高,远超一般棋类。每回合中,玩家不仅要考虑移动策略,还需规划绕行路径,这使得该游戏在计算机博弈中状态空间庞大,博弈树复杂度极高。因此,设计适合苏拉卡尔塔棋的高效搜索算法成为计算机博弈的关键难点。由于其特殊规则和高复杂度,苏拉卡尔塔棋在机器博弈研究中具有独特的应用价值。通过研究该棋种,研究者可探索复杂局面下的搜索优化方法与策略规划。近年来, NegaScout (负极大剪枝算法) 等算法被引入苏拉卡尔塔棋博弈中,以提升计算机的智能决策能力。

## 3 苏拉卡尔塔棋博弈

### 3.1 苏拉卡尔塔棋的棋盘设计

苏拉卡尔塔棋的棋盘设计具有独特性,主要表现为一个  $6 \times 6$  的网格布局,并结合了四个角落的弧形路径。这种棋盘布局不仅提供了传统棋类的基本方格结构,还允许棋子沿弧形路径移动,从而增添了游戏的策略性和复杂性。为了方便玩家更直观地理解棋盘,设计中在棋盘外部添加了坐标标识。横向使用字母 A—F 标识列,纵向使用数字 1—6 标识行,使玩家在记录和分析棋局时更加便捷。此外,棋盘格子通过清晰的线条分割,增强了可视性,使每个棋子的放置位置更明确。

## 3.2 评估函数

### 3.2.1 初始化得分变量

函数中定义了多个变量以存储不同得分要素,包括了用于分别存储黑棋和红棋总得分的变量。此外,还定义了多个变量,用于分别记录黑棋和红棋的攻击和防守价值。

### 3.2.2 攻击与防守价值的计算

在正式进行得分计算之前,调用了两个辅助函数,用于分别计算棋盘上每个棋子的攻击和防守价值。这两个函数通过遍历棋盘并根据棋子的特定位置和周围棋子的关系(例如棋子之间的相对位置和颜色),来决定每个位置的攻击或防守分值。

### 3.2.3 棋盘遍历和得分计算

评估函数嵌套了两个 for 循环,以遍历整个  $6 \times 6$  的棋盘,并根据棋子的颜色(1 表示黑棋,2 表示红棋)进行不同的分值处理。每个棋子的基础得分为 100 分,若棋子位于对角线上,则额外增加 20 分。除此之外,根据攻击和防守值的不同(如攻击值大于等于 100,则扣除 100 分),进一步调整双方的总得分。

### 3.2.4 回合调整

评估函数通过参数来判断是否为黑棋的回合,并根据该信息调整得分。在黑棋回合时,如果棋盘上某红棋的攻击价值大于防御价值,则黑棋得分加 100 分;反之,则对红棋得分进行类似处理。通过这种回合依赖的动态调整,使得评估函数能够更准确地评估当前回合的优势。

### 3.2.5 最终得分计算与输出

在遍历完棋盘并计算每个位置的得分后,计算最终的黑棋和红棋总得分。若为黑棋回合,则返回黑棋、红棋的分值差异,表示黑棋优势程度;否则,返回红棋、黑棋的分值差异。

## 3.3 NegaScout 算法

### 3.3.1 NegaScout 算法的机制

NegaScout 算法是一种基于 Alpha-Beta 剪枝的优化搜索算法,旨在减少博弈树搜索中的冗余计算,提高整体搜索效率。它的核心机制是通过一个逐步收缩的搜索窗口来动态调整搜索范围,从而避免对不必要的节点进行完全搜索。

NegaScout 算法在每个节点的搜索过程中,首先对第一个子节点使用一个完整的搜索窗口,这个过程类似于标准的 Alpha-Beta 剪枝。在获取第一个子节点的评估值后,算法对接下来的每一个子节点采用一个较小的紧缩窗口(即 Null Window)进行搜索,以确定是否存在超出当前已知界限的得分。如果紧缩窗口的搜索结果表明评估值超出了已知的上下限, NegaScout 会重新对该节点进行一次完整窗口的搜索,从而获得一个准确的评估结果。这种重新搜索的机制有效地平衡了搜索的效率与评估结果的精度。通过初始的紧缩窗口搜索,算法能够在多数情况下排除无关的子节点,而不必耗

费时间对每个节点进行完整搜索。当确实有必要时，才重新扩大搜索窗口来获取准确的评估值，这种设计显著减少了冗余计算量。NegaScout 算法的优势在于它能够高效处理具有大量节点的深度搜索问题。相比传统的 Alpha-Beta 剪枝，NegaScout 通过对后续子节点的紧缩搜索，大幅减少了无效节点的完全探索，从而提高了整体的搜索效率。特别是在处理分支庞大的博弈树时，NegaScout 的性能优势尤为明显。此外，NegaScout 算法通常结合历史得分和哈希表等优化策略。历史得分表是基于之前搜索过程中积累的经验数据，帮助算法优先探索效果较好的走法；而哈希表用于缓存之前已计算过的节点，以减少重复计算的可能性。这些优化进一步提升了算法的执行效率，使得 NegaScout 在复杂博弈搜索中表现出色。

### 3.3.2 NegaScout 算法的实现

在每个节点的搜索过程中，我们会首先检查当前局面是否已达到终局状态，如果发现游戏已经结束，则直接返回终局的评估得分。接下来，根据当前的深度确定执行着法的棋方，并在哈希表中查找当前局面的得分，如果找到符合条件的结果，则直接返回该得分，避免重复计算。当搜索深度为零时，我们会视该节点为叶子节点，调用评估函数对局面进行打分，并将结果存入哈希表中。然后，算法会生成所有可能的走法，并通过历史得分对这些走法进行排序，从而优先处理得分较高的走法。对于每一个可能的走法，NegaScout 算法都会模拟该走法，并在局面上执行递归搜索。在第一个走法之后，NegaScout 算法会使用一个较小的搜索窗口进行紧缩搜索，并在某些条件下进行重新搜索，以确保评估结果的准确性。在每次模拟完成后，NegaScout 算法都会撤销之前的操作，恢复局面状态。若某个节点的得分超出预设的上限，NegaScout 算法会立即进行剪枝操作，停止对该节点的进一步搜索。最后，NegaScout 算法将所有可能走法的最佳结果记录在哈希表中，根据具体的评估情况选择是否存储为精确值或上限值，并返回当前节点的最佳得分。

## 4 实验结果与分析

为了评估 NegaScout 算法在博弈树搜索中的实际性能表现，我们将 NegaScout 算法与传统的 Alpha-Beta 剪枝算法进行了对比实验。实验的主要目标是通过测量两种算法在相同局面和相同搜索深度下的搜索效率和效果，以验证

NegaScout 算法在减少冗余节点、提高搜索效率方面的优势。实验结果如表 1 所示，展示了两种算法在不同搜索深度下的节点访问数量和搜索时间。

表 1 NegaScout 与 Alpha-Beta 搜索速度对照

深度	Alpha-Beta 剪枝算法 搜索时间 /s	NegaScout 算法搜索 时间 /s
2	0.8	0.5
4	3.2	2.0
6	12.8	8.0

从表 1 中可以看出，NegaScout 算法在不同的搜索深度下搜索效率均显著快于 Alpha-Beta 剪枝算法，并且在相同深度下，NegaScout 的搜索时间明显少于 Alpha-Beta 剪枝算法。这表明 NegaScout 算法通过紧缩窗口搜索和重新搜索的策略，成功减少了博弈树中的冗余节点，提高了搜索效率。

## 5 结语

机器博弈是一个富有挑战和探索潜力的研究领域，同时也蕴含着极大的乐趣。在苏拉卡尔塔棋的计算机博弈中，关于博弈策略的研究与学习依然有待深入，许多未知的技术仍等待着被发掘与应用。论文对当前苏拉卡尔塔棋机器博弈中较为常见的两种搜索策略，即基于“NegaScout 算法”和基于“Alpha-Beta 剪枝算法”进行了详细的介绍和对比研究。文章设计并实现了 NegaScout 算法策略，显著提升了搜索效率并减少了决策时间，从而增强了程序的棋力水平。这一改进为苏拉卡尔塔棋智能对弈提供了新的技术支持。

### 参考文献

- [1] 张涛,江业峰,李博文.基于PVS算法的苏拉卡尔塔棋博弈系统设计与实现[J].信息与电脑(理论版),2023,35(19):46-48.
- [2] 李东轩,胡伟,王静文.基于Alpha-Beta算法的苏拉卡尔塔棋博弈系统研究[J].智能计算机与应用,2022,12(2):123-125.
- [3] 车晓菲,徐勇,蒋宗华.苏拉卡尔塔棋系统的设计与实现[J].信息与电脑(理论版),2021,33(6):70-73.
- [4] 李淑琴,李静波,韩裕华,等.苏拉卡尔塔棋博弈系统中评估函数的研究[J].北京信息科技大学学报(自然科学版),2012,27(6):42-45+61.
- [5] 王仁泉,丁濛,李淑琴,等.基于强化学习的苏拉卡尔塔棋博弈算法[J].智能计算机与应用,2020,10(4):6-8+12.