



(12)发明专利申请

(10)申请公布号 CN 105817029 A

(43)申请公布日 2016.08.03

(21)申请号 201610145217.0

(22)申请日 2016.03.14

(71)申请人 安徽大学

地址 230601 安徽省合肥市经济技术开发区
九龙路111号

(72)发明人 李学俊 汪坤兵 朱二周 吴蕾
李龙澍

(74)专利代理机构 南京知识律师事务所 32207
代理人 高玲玲

(51)Int.Cl.

A63F 11/00(2006.01)

G06F 17/30(2006.01)

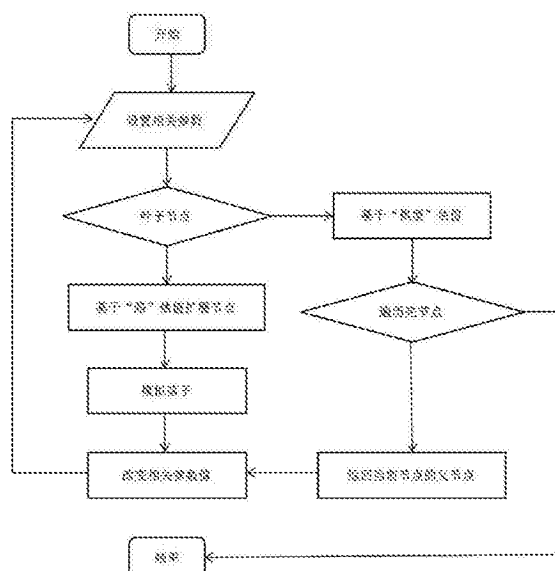
权利要求书2页 说明书8页 附图1页

(54)发明名称

六子棋博弈系统中基于路和棋型的混合搜索方法

(57)摘要

本发明公开了一种六子棋博弈系统中基于路和棋型的混合搜索方法,本发明是在使用Alpha-Beta剪枝算法对博弈树进行搜索时扩展中间节点阶段对候选节点的估值使用了基于“路”的估值方式,在对叶子节点进行估值时使用了基于“棋型”的估值方式。将这两种估值方式混合使用,应用于Alpha-Beta剪枝搜索中,结合两者的优点。混合搜索模式与单一的基于“路”的搜索模式相比,在搜索效率相同的情况下提高了博弈水平,与单一的基于“棋型”的搜索模式相比,在相同的博弈水平下提高了搜索效率。



1. 一种六子棋博弈系统中基于路和棋型的混合搜索方法,其特征在于包括以下步骤:

步骤一:设置节点信息、宽度值、深度值、Alpha值、Beta值和玩家;

步骤二:判断是否为叶子节点,如果为叶子节点,则执行步骤六,否则执行步骤三;

步骤三:扩展节点,获取所有待扩展的节点,使用基于“路”的方式对所有待扩展的节点进行估值,并按照设置的宽度选取节点数,作为扩展的节点返回;

步骤四:根据搜索结果模拟落子;

步骤五:将深度减一,调换Alpha值和Beta值,并改变玩家,然后执行步骤一;

步骤六:采用基于“棋型”的方式对叶子节点进行估值,然后返回到当前节点的父节点;

步骤七:判断当前节点的深度是否为设置的深度,即是否为根节点,当遍历完根节点的孩子节点时,搜索结束。

2. 根据权利要求1中所述的混合搜索方法,其特征在于:所述步骤三基于“路”方式进行估值采用下式的估值函数:

$$\begin{aligned} \text{TotalScore} = & \sum_{k=0}^6 (\text{NumberOfMyRode}[k] * \text{ScoreOfMyRode}[k]) \\ & - \sum_{k=0}^6 (\text{NumberOfThreatRode}[k] * \text{ScoreOfThreatRode}[k]) + \text{PosValueOfNode} \quad (1) \end{aligned}$$

其中,TotalScore:代表当前局面的总估值;NumberOfMyRode:代表当前局面我方每路的路数;ScoreOfMyRode:代表我方每路的价值;NumberOfThreatRode:代表当前棋局对方各路的路数;ScoreOfThreatRode:代表对方各路的价值;PosValueOfNode:代表当前走法中两个落子在棋盘上的位置值。

3. 根据权利要求1中所述的混合搜索方法,其特征在于:所述步骤三基于“棋型”方式进行估值采用下式的估值函数:

$$\text{TotalScore} = \text{ScoreOfMy} + \text{ScoreOfThreat} + \text{ScoreOfExtra} + W \quad (2)$$

$$\text{ScoreOfMy} = \text{FirstStoneOfMy} + \text{SecondStoneOfMy} + \text{AddPowOfMy} \quad (3)$$

$$\text{ScoreOfThreat} = \text{FirstStoneOfThreat} + \text{SecondStoneOfThreat} + \text{AddPowOfThreat} \quad (4)$$

其中,ScoreOfMy:代表我方当前局面的估值;ScoreOfThreat:代表对方对我方产生的威胁值;ScoreOfExtra:代表额外的综合估值;W:代表比重失衡因子;FirstStoneOfMy:代表我方第一颗落子的估值;SecondStoneOfMy:代表我方第二颗落子的估值;AddPowOfMy:代表我方两颗落子关联方向的附加值;FirstStoneOfThreat:代表模拟对方第一颗落子对方产生的威胁值;SecondStoneOfThreat:代表模拟对方第二颗落子对我方产生的威胁值;AddPowOfThreat:代表模拟对方关联方向上对我方产生的威胁值。

4. 根据权利要求1中所述的混合搜索方法,其特征在于:所述比重失衡因子W的设置采用下式:

$$W = W1 + W2 \quad (5)$$

$$K1 = \max1 / \min1 \quad (6)$$

$$K2 = \max2 / \min2 \quad (7)$$

$$W1 = \begin{cases} -\min1/2 & k1 \geq 2.5 \\ 0 & k1 < 2.5 \end{cases} \quad (8)$$

$$W2 = \begin{cases} -\min2/2 & k2 \geq 2.5 \\ 0 & k2 < 2.5 \end{cases} \quad (9)$$

其中:W1:代表我方第一颗落子估值(FirstStoneOfMy)与对方第一颗落子估值(FirstStoneOfThreat)失衡时产生的差;W2:代表我方第二颗落子估值(SecondStoneOfMy)与对方第二颗落子估值(SecondStoneOfThreat)失衡时产生的差值;K1:代表第一颗棋子估值的失衡因子;K2:代表第二颗棋子估值的失衡因子;max1和min1:分别代表我方和对方第一颗棋子的估值的大者和小者;max2和min2:分别代表我方和对方第二颗棋子的估值的大者和小者。

六子棋博弈系统中基于路和棋型的混合搜索方法

技术领域

[0001] 本发明属于棋类游戏机器博弈的研究领域,特别涉及一种六子棋博弈系统中基于路和棋型的混合搜索方法。

背景技术

[0002] 机器博弈是人工智能的主要研究领域之一,就是利用计算机实现人机对战或是机机对战,使计算机能够模仿人的思维过程。博弈论从开始到现在,在国内外得到迅速发展,很多学者和专家在博弈领域做出贡献,曾经超级计算机Hydra获胜英国象棋冠军迈克尔·亚当斯。因此,在规定的时间内提高博弈水平是博弈的最终目标。但是由于受到硬件系统和六子棋本身复杂度的限制,要想提高博弈水平,只有从博弈树的搜索和估值两方面考虑。

[0003] 六子棋英文名connect6,来源于五子棋,是由我们台湾国立交通大学资讯工程系教授吴毅成提出。因六子棋跟五子棋相比,到目前为止更具有公平性,规则简单,玩法复杂等特点。2006年成为国际奥林匹克比赛项目,2007年首次成为我国大学生机器博弈锦标赛比赛项目。六子棋比赛规则是黑方先下一颗棋子,然后每方轮流下两颗棋子,先连成六连者为胜,没有禁手限制,若下面棋盘仍未分出胜负,为和棋。但有时也可另加规则,当和棋时,通过判断形成五连的多少来决定胜负,从而减少和棋的状况。六子棋的复杂性体现在19x19的棋盘和对棋型的判断,棋盘上存在361个点,导致博弈树分支因子过大;棋型有11种,包括六连,活五,眠五,活四,眠四,活三,朦胧三,眠三,活二,眠二和单一,其中每种棋型又分为多种情况,因此它的状态空间复杂度可以达到10的172次方,复杂度相当于围棋;同时棋局不仅多,而且变化比较复杂。所以六子棋本身的特性和难度,更吸引无数爱好者进行研究,其中主要包括两大部分,估值策略和搜索算法,由于存在难度,目前估值策略大部分使用静态估值,搜索算法使用极大极小算法和AlphaBeta剪枝算法。这些都导致博弈树搜索效率低,估值不够准确,从而影响机器博弈水平。

[0004] 六子棋的棋盘相对于其它棋种来说比较大,节点比较多,棋局的估值分析也比较复杂。通常应用于六子棋的博弈算法是Alpha-Beta剪枝算法,其来源于极大极小算法,虽然使用了剪枝,在一定程度上提高了搜索效率,但是博弈水平效果仍不明显。当对博弈树进行搜索时该算法搜索的节点数会随搜索深度的递增成指数型增长,而搜索效率会明显降低,其原因除了节点的指数增长之外,一个很大的影响因数是在扩展节点时先由走法生成器获得所有合法走法,然后对所有走法进行估值的方式,即存在于伪码GetNMove获取扩展节点函数中。主要存在两种估值方式,一种是基于路的估值,另一种是基于棋型的估值,各有优缺点。在搜索技术上,目前使用的仍然是单一的搜索方式,因此Alpha-Beta搜索算法不能同时达到高效率高水平的程度。

[0005] 因此,在目前的六子棋博弈软件中,当使用AlphaBeta剪枝算法搜索博弈树时使用的都是单一估值方式的搜索,即搜索过程中在对中间节点和叶子节点使用的都是同样的估值方式,要么都是基于“路”的,要么都是基于“棋型”的。由已完成得实验可得:如果只采用单一的基于“路”的Alpha-Beta剪枝搜索,搜索效率要高,但是棋局估值准确度不高;如果只

采用单一的基于“棋型”的Alpha-Beta剪枝搜索,估值比较准确,但是搜索效率受到很大限制。所以不管使用哪种方式,都会存在不足,因此需要对此进行优化。

发明内容

[0006] 本发明的目的是针对现有技术的缺陷,提供一种六子棋博弈系统中基于路和棋型的混合搜索方法。

[0007] 为了实现上述目的,本发明采用以下技术方案:一种六子棋博弈系统中基于路和棋型的混合搜索方法,包括以下步骤:

- 步骤一:设置节点信息、宽度值、深度值、Alpha值、Beta值和玩家;
- 步骤二:判断是否为叶子节点,如果为叶子节点,则执行步骤六,否则执行步骤三;
- 步骤三:扩展节点,获取所有待扩展的节点,使用基于“路”的方式对所有待扩展的节点进行估值,并按照设置的宽度选取节点数,作为扩展的节点返回;
- 步骤四:根据搜索结果模拟落子;
- 步骤五:将深度减一,调换Alpha值和Beta值,并改变玩家,然后执行步骤一;
- 步骤六:采用基于“棋型”的方式对叶子节点进行估值,然后返回到当前节点的父节点;
- 步骤七:判断当前节点的深度是否为设置的深度,即是否为根节点,当遍历完根节点的子节点时,搜索结束。

[0008] 所述步骤三基于“路”方式进行估值采用下式的估值函数:

$$\text{TotalScore} = \sum_{k=0}^6 (\text{NumberOfMyRode}[k] * \text{ScoreOfMyRode}[k]) - \sum_{k=0}^6 (\text{NumberOfThreatRode}[k] * \text{ScoreOfThreatRode}[k]) + \text{PosValueOfNode} \quad (1)$$

其中,TotalScore:代表当前局面的总估值;NumberOfMyRode:代表当前局面我方每路的路数;ScoreOfMyRode:代表我方每路的价值;NumberOfThreatRode:代表当前棋局对方各路的路数;ScoreOfThreatRode:代表对方各路的价值;PosValueOfNode:代表当前走法中两个落子在棋盘上的位置值。

[0009] 所述步骤三基于“棋型”方式进行估值采用下式的估值函数:

$$\text{TotalScore} = \text{ScoreOfMy} + \text{ScoreOfThreat} + \text{ScoreOfExtra} + W \quad (2)$$

$$\text{ScoreOfMy} = \text{FirstStoneOfMy} + \text{SecondStoneOfMy} + \text{AddPowOfMy} \quad (3)$$

$$\text{ScoreOfThreat} = \text{FirstStoneOfThreat} + \text{SecondStoneOfThreat} + \text{AddPowOfThreat} \quad (4)$$

其中,ScoreOfMy:代表我方当前局面的估值;ScoreOfThreat:代表对方对我方产生的威胁值;ScoreOfExtra:代表额外的综合估值;W:代表比重失衡因子;FirstStoneOfMy:代表我方第一颗落子的估值;SecondStoneOfMy:代表我方第二颗落子的估值;AddPowOfMy:代表我方两颗落子关联方向的附加值;FirstStoneOfThreat:代表模拟对方第一颗落子对方产生的威胁值;SecondStoneOfThreat:代表模拟对方第二颗落子对我方产生的威胁值;AddPowOfThreat:代表模拟对方关联方向上对我方产生的威胁值。

[0010] 所述比重失衡因子W的设置采用下式:

$$W = W1 + W2 \quad (5)$$

$$K1 = \max1 / \min1 \quad (6)$$

$$K2 = \max2 / \min2 \quad (7)$$

$$W1 = \begin{cases} -\min1/2 & k1 \geq 2.5 \\ 0 & k1 < 2.5 \end{cases} \quad (8)$$

$$W2 = \begin{cases} -\min2/2 & k2 \geq 2.5 \\ 0 & k2 < 2.5 \end{cases} \quad (9)$$

其中:W1:代表我方第一颗落子估值(FirstStoneOfMy)与对方第一颗落子估值(FirstStoneOfThreat)失衡时产生的差;W2:代表我方第二颗落子估值(SecondStoneOfMy)与对方第二颗落子估值(SecondStoneOfThreat)失衡时产生的差值;K1:代表第一颗棋子估值的失衡因子;K2:代表第二颗棋子估值的失衡因子;max1和min1:分别代表我方和对方第一颗棋子的估值的大者和小者;max2和min2:分别代表我方和对方第二颗棋子的估值的大者和小者。

[0011] 在Alpha-Beta剪枝搜索过程中,对于中间待扩展的节点的估值和叶子节点的估值,可以存在四种混合方式:

(1)两次都使用基于“路”的方式。

[0012] (2)两次都使用基于“棋型”的方式。

[0013] (3)先使用“棋型”,后使用“路”。

[0014] (4)先使用“路”,后使用“棋型”。

[0015] 对于第一种方式,局面估值缺乏准确性,导致搜索不够准确。第二种方式虽然估值准确,但是搜索效率很低,深度为4时可以接受,深度为6时就不可接受了。对于第三种方式来说,更是不可取的交叉方式。最后一种是本发明所采用的交叉方式,结合了“路”和“棋型”的搜索效率和搜索准确性的双重优势,搜索深度可以到10,甚至到20,宽度也可以设置的更大,这样搜索效率得到保证,并且博弈水平明显提高。

[0016] 本发明是在使用Alpha-Beta剪枝算法对博弈树进行搜索时扩展中间节点阶段对候选节点的估值使用了基于“路”的估值方式,在对叶子节点进行估值时使用了基于“棋型”的估值方式。将这两种估值方式混合使用,应用于Alpha-Beta剪枝搜索中,结合两者的优点。在整个搜索过程中将基于路的估值方式和基于棋型的估值方式进行了混合顺序使用,各自在相应的搜索阶段显示出自己的优点,从总体上均衡搜索效率和博弈水平。混合搜索模式与单一的基于“路”的搜索模式相比,在搜索效率相同的情况下提高了博弈水平,与单一的基于“棋型”的搜索模式相比,在相同的博弈水平下提高了搜索效率。

[0017] 最后在Ubuntu系统下,以QTcreator为开发工具,设计了的六子棋博弈系统软件。并将基于路和基于棋型的混合估值模式应用于该系统中,提高系统的搜索效率和博弈水平。

附图说明

[0018] 图1是混合搜索方法的流程图。

具体实施方式

[0019] 为了更好地理解本发明的技术方案,以下结合附图对本方案进行更加详细的描述。

[0020] 步骤一:设置节点信息、宽度值、深度值、Alpha值、Beta值和玩家;

步骤二:判断是否为叶子节点,如果为叶子节点,则执行步骤六,否则执行步骤三;

步骤三:扩展节点,获取所有待扩展的节点,使用基于“路”的方式对所有待扩展的节点进行估值,并按照设置的宽度选取节点数,作为扩展的节点返回;基于“路”方式进行估值采用下式的估值函数:

$$\text{TotalScore} = \sum_{k=0}^6 (\text{NumberOfMyRode}[k] * \text{ScoreOfMyRode}[k]) - \sum_{k=0}^6 (\text{NumberOfThreatRode}[k] * \text{ScoreOfThreatRode}[k]) + \text{PosValueOfNode}$$

TotalScore:代表当前局面的总估值,

NumberOfMyRode:代表当前局面我方每路的路数,

ScoreOfMyRode:代表我方每路的价值,

NumberOfThreatRode:代表当前棋局对方各路的路数,

ScoreOfThreatRode:代表对方各路的价值,

PosValueOfNode:代表当前走法中两个落子在棋盘上的位置值。

[0021] 整个局面的估值为我方估值减去对方估值再加上棋盘位置值,当对我方有利时我方估值比较大,对对方不利时,对方估值较小,这样总的估值就会变大,就说明当前的走法比较好。这种基于“路”的估值方式特征就是估值函数比较简单,同时又不需要对棋子进行复杂的分析,所以估值效率比较高。按照设定的宽度值选取节点,作为扩展的节点返回。

[0022] 步骤四:根据搜索结果模拟落子;

步骤五:将深度减一,调换Alpha值和Beta值,并改变玩家,然后执行步骤一;

步骤六:采用基于“棋型”的方式对叶子节点进行估值,然后返回到当前节点的父节点;当对叶子节点进行估值时,使用的是基于“棋型”的估值方式,局面的估值函数如下:

$$\text{TotalScore} = \text{ScoreOfMy} + \text{ScoreOfThreat} + \text{ScoreOfExtra} + W$$

ScoreOfMy:代表我方当前局面的估值,

ScoreOfThreat:代表对方对我方产生的威胁值,

ScoreOfExtra:代表额外的综合估值,

W:代表比重失衡因子。

[0023] 该估值函数的估值趋势就是选取既对我方有利又可以消除对方对我方形成的威胁,即使ScoreOfMy和ScoreOfThreat同时达到比较大的值,并通过对后项的综合分析考虑,从总体上进行评估。其中ScoreOfMy和ScoreOfThreat的估值包括如下公式:

$$\text{ScoreOfMy} = \text{FirstStoneOfMy} + \text{SecondStoneOfMy} + \text{AddPowOfMy}$$

$$\text{ScoreOfThreat} = \text{FirstStoneOfThreat} + \text{SecondStoneOfThreat} + \text{AddPowOfThreat}$$

FirstStoneOfMy:代表我方第一颗落子的估值,

SecondStoneOfMy:代表我方第二颗落子的估值,

AddPowOfMy:代表我方两颗落子关联方向的附加值,

FirstStoneOfThreat:代表模拟对方第一颗落子对方产生的威胁值,

SecondStoneOfThreat:代表模拟对方第二颗落子对我方产生的威胁值,

AddPowOfThreat:代表模拟对方关联方向上对我方产生的威胁值。

[0024] 基于“棋型”的估值需要落子进行棋型分析,包括第一颗落子和第二颗落子,然后是对每个落子的四个方向进行扫描判断棋型(除关联方向的扫描),如果两个落子存在关联方向,需要单独对关联方向进行信息统计,然后再进行估值。关联是指两颗落子同时存在于水平,垂直,左斜和右斜的四个方向之一上,并且距离小于等于5。

[0025] 对于ScoreOfExtra的值,其中包括产生的迫著数,边界干扰和可扩展空间的大小。这些不同的因素都会对总的估值产生影响。对于有迫著的,可以依次判断迫著数,例如迫著数大于等于3时,已找到必赢的走法,将估值调整到极大值。对于边界干扰,原则是越靠近边界,是不利的位置,就要适当减小估值。对于可扩展空间的分析是当位于空旷位置时需要增加估值,如果两个落子之间空位很少,那么就不具有延展性。

[0026] 对于W值的考虑,公式如下:

$$W=W1+W2$$

$$K1=\max1/\min1$$

$$K2=\max2/\min2$$

$$W1=\begin{cases} -\min1/2 & k1 \geq 2.5 \\ 0 & k1 < 2.5 \end{cases}$$

$$W2=\begin{cases} -\min2/2 & k2 \geq 2.5 \\ 0 & k2 < 2.5 \end{cases}$$

W1:代表我方第一颗落子估值(FirstStoneOfMy)与对方第一颗落子估值(FirstStoneOfThreat)失衡时产生的差值,

W2:代表我方第二颗落子估值(SecondStoneOfMy)与对方第二颗落子估值(SecondStoneOfThreat)失衡时产生的差值,

K1:代表第一颗棋子估值的失衡因子,

K2:代表第二颗棋子估值的失衡因子,

max1和min1:分别代表我方和对方第一颗棋子的估值的大者和小者,

max2和min2:分别代表我方和对方第二颗棋子的估值的大者和小者。

[0027] 当失衡因子大于等于2.5时,说明模拟我方第一颗落子的估值和对方第一颗落子的威胁值相差较大,即当前落子只对一方有利,要么是只具有进攻没有防守,要么是只具有防守没有进攻,当前走法并不是所需要的,所以需要适当降低估值。

[0028] 基于“棋型”的估值函数需要考虑很多因素,函数也比较复杂,同时还需要对棋子进行复杂的分析,棋型的判断,所以估值效率较低,但是估值准确性很高。

[0029] 步骤七:判断当前节点的深度是否为设置的深度,即是否为根节点,当遍历完根节

点的子节点时,搜索结束。

[0030] 在搜索的初期(扩展节点)时,由于在深度搜索的初期对局面的估值准确性要求不高,不管怎么估值,只要假设能够保证最佳走法落在设置的宽度范围内,在经过最后的回溯,就能确定最佳点。因此需要设置比较大的宽度和深度,而在此时,如果只使用单一的基于“棋型”的估值方式是远远满足不了这个需求,因为棋型的分析和估值比较复杂,搜索效率低。所以另一种基于“路”的估值方式就可以满足这个要求,并且基于“路”的估值也是可以保证一定的准确性的,因此,在不失估值准确性下应该使用基于“路”方式对待扩展的节点进行估值,可以使搜索的宽度和深度增大。

[0031] 当搜索到叶子节点时,由于已达到预先设置的深度,因为六子棋的棋盘很大,不可能将棋局模拟结束(即分出胜负的状态),所以只能模拟到一定的深度,然后对局面进行估值,并以此估值代表最终的模拟结果,该结果肯定没有模拟分出胜负的结果准确,所以叶子节点的估值准确性至关重要。如果此时仍然采用基于“路”的方式对叶子节点估值,是无法保证很高的准确性。那么就应该采用基于“棋型”的方式对棋局进行估值,此时对搜索效率要求并不是很高,采用基于棋型的估值方式进行搜索,可以提高搜索的准确性。

[0032] 在回溯过程中,仍然采用剪枝方式。在整个搜索过程中将基于路的估值方式和基于棋型的估值方式进行了混合顺序使用,各自在相应的搜索阶段显示出自己的优点,从总体上均衡搜索效率和博弈水平。

[0033] 混合搜索模式与单一的基于“路”的搜索模式相比,在搜索效率相同的情况下提高了博弈水平,与单一的基于“棋型”的搜索模式相比,在相同的博弈水平下提高了搜索效率。

[0034] 系统设计

将基于“路”的估值与基于“棋型”的估值分别用于Alpha-Beta搜索中对六子棋博弈树待扩展节点和叶子节点的估值,在此基础上设计了六子棋博弈平台。主要包括以下几个模块:

- (1) 棋盘表示
- (2) 走法生成
- (3) 估值函数
- (4) 搜索
- (5) 开局库
- (6) 界面

棋盘使用二维数组表示;走法生成模块用于生成所有合法的走法;估值函数模块包括基于“路”的估值函数和基于“棋型”的估值函数;搜索模块包括模拟连续迫著和Alpha-Beta剪枝搜索;开局库用于开局局面的匹配;系统的执行流程:首先是输入走法,然后判断是否为开局阶段,如果是就进行不做深度的估值和搜索,因为开局做深度搜索无意义,如果不是开局,就进行模拟连续迫著,连续迫著是用于判断是否能够找到必赢的走法,如果迫著成功,将输出走法,如果不成功,再进行防守型Alpha-Beta剪枝深度搜索,输出最佳走法。

[0035] 基于“路”的估值函数

在基于“路”的估值函数中,首先需要解决的是各路价值的设定。一般的设置是增长式的设置,即路数越大,价值越大,比如1路可以为30,2路为90,3路为200,4路为800,5路为1000,6路为必赢,应设置为极大值如1000000。

[0036] 接着是棋子在棋盘中的位置值的设置,根据经验可知,棋盘中心位置为开局最佳位置,离中心点越近,价值越高,比如可以设置中心位置值为10,然后沿四个方向逐层减一。

[0037] 最后就是对棋盘进行扫描,统计我方和对方棋子各路的数量。为了避免无效的扫描,只统计有效的路数,并同时应该缩小扫描的范围,即缩小棋盘,使能够用最小棋盘包含棋盘上所有的棋子。

[0038] 基于“棋型”的估值函数

在该估值函数中对单个落子点进行估值时,将从四个方向依次进行估值,那么具体的扫描方式是以包含落子点距离为5的范围内进行扫描,然后判断出相应的棋型。而对于棋型的判断方式则是先对该方向进行信息统计,分析,然后根据相邻棋子的连接情况分别判断,最后给出匹配的一种棋型。对于价值表的设置是存在两张估值表,例如以下两张表:

表一:

棋型	价值
眠一	20
单关	30
眠二	510
跳二	900
活二	1000
眠三	1400
朦胧三	2800
活三	3000
眠四	7000
活四	12000
眠五	500
活五	1000
活六	100000

表二:

棋型	价值
眠一	10
单关	20
眠二	200
跳二	450
活二	600
眠三	1300
朦胧三	2100
活三	2400
眠四	5500
活四	9000
眠五	8000
活五	10000

活六	300000
----	--------

表1是我方各棋型的估值,表2是对方棋型的威胁值,当判断出棋型之后,可以依据此表返回相应的估值。如果两颗棋子存在关联,还需要对关联方向上的信息进行记录和分析,然后给出棋型判断和附加值。

[0039] 对于估值函数中失衡因子大小的确定,本实实例采用的是2.5,可以根据自己的经验设置该值,也可以通过实验选取恰当的值。

[0040] 棋盘表示

首先六子棋的棋盘是19x19的矩形,每一个交叉点就是一颗棋子的位置,这里是用一个二维数组chess[19][19]来表示棋子在棋盘上的位置,初始值都为0。对于棋子的颜色,这里使用的是一个整型数,1代表黑方,7代表白方。例如当chess[1][2]等于1时,代表该位置上是黑棋。落子即将该位置的初始值0赋予相应的1或7以上所述,仅是本发明的较佳实施例而已,并非对本发明做任何形式的限制。凡是依据本发明的技术和方法实质对以上实施例所作的任何简单修改、等同变化与修饰,均仍属于本发明的技术和方法方案的范围内。

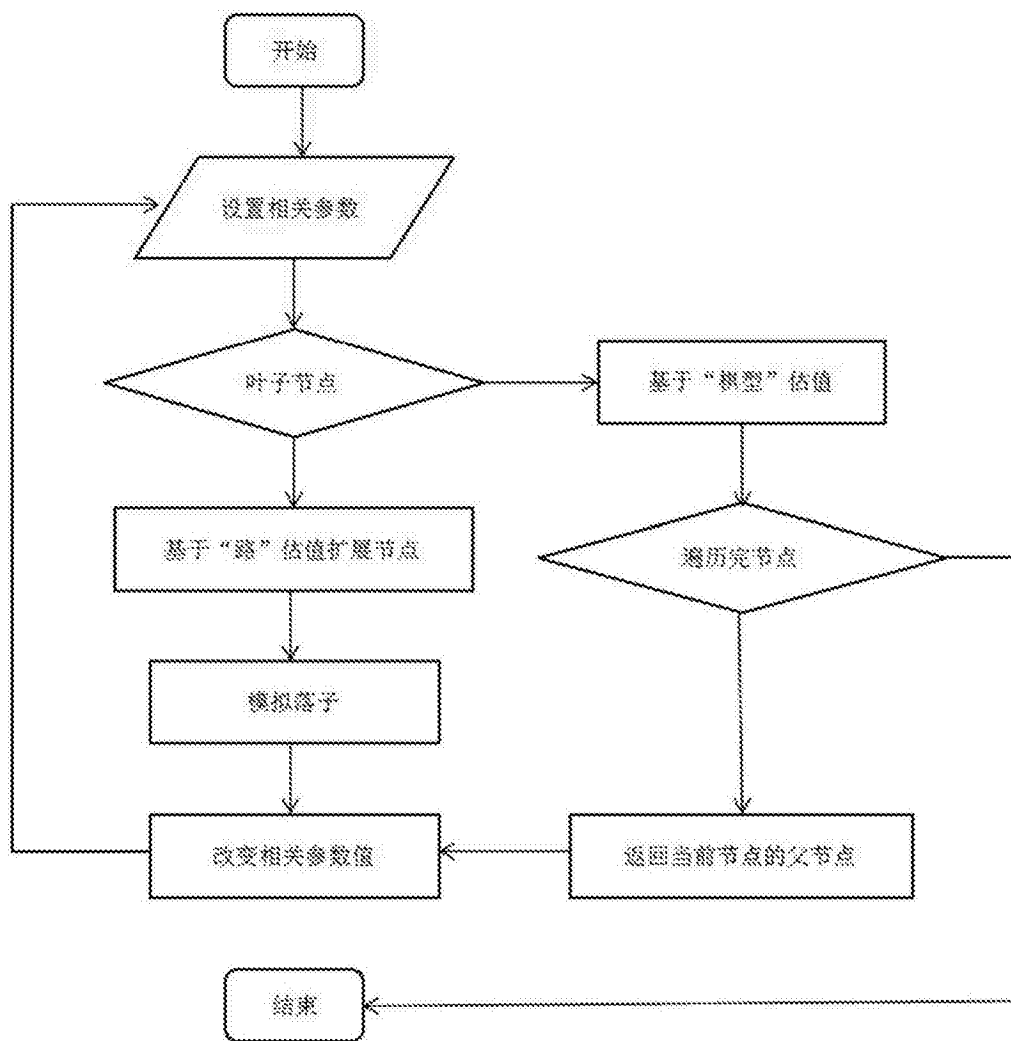


图1