

中国象棋机器博弈的时间自适应分配策略研究

徐长明,南晓斐,王 骄,徐心和

(东北大学 人工智能与机器人研究所,辽宁 沈阳 110004)

摘要:时间分配问题在象棋比赛中是十分重要的策略问题,在机器博弈中也是如此。好的策略可以把宝贵的时间资源用在“刀刃”上;此外,好的时间分配策略还要有好的自适应性,亦即对大多数比赛,无论其限时的长短、步数的多寡,该策略都能合理利用时间。在分析和建立了时间分配的数学模型的基础上,介绍了自适应时间分配与调整的策略和算法。时间分配问题与搜索和评估密切相关,也影响着机器博弈的风格。

关键词:机器博弈;时间分配策略;自适应

中图分类号:TP18 文献标识码:A 文章编号:1673-4785(2006)02-0039-05

Adaptive time allocation strategy in computer game of Chinese Chess

XU Chang-ming, NAN Xiao-fei, WANG Jiao, XU Xin-he

(Institute of Artificial Intelligence and Robotics, Northeastern University, Shenyang 110004, China)

Abstract: Time Allocation is a typical and important strategy problem in chess matches, so does in computer games. A good strategy can allocate time for each searching-step to make the best use of it. Furthermore, a good allocation strategy should be adaptive, namely, fitting most contests, regardless of the length of time limit, or the number limit of steps. Based on analysing and establishing the mathematic model of time allocation, this paper introduces an adaptive algorithm of time allocation and adjustment. The method ties up with the searching and evaluation module, and also affects the style of computer games.

Key words: computer games; time allocation strategy; adaptive

正式比赛过程中,时钟是不可或缺的。每走完一步都要按时钟,将计时切换到对手一方。时间是参赛选手的宝贵资源,也是杀手,因时间没有利用好而输棋屡见不鲜。例如,优势情况下因为超时而输掉比赛,或者输了棋还剩余很多时间,均说明没有很好地利用时间资源。因此,深入研究中国象棋机器博弈的时间分配问题^[1]很有意义,对其他棋类博弈也应有借鉴意义。

时间的分配策略可以分为2大类:静态分配策略和动态分配策略。首先介绍时间分配问题的数学模型,然后介绍了几种简单的静态分配策略,分析其优缺点,接着介绍时间的动态分配策略。最后,在“平均配时”模型的基础上,重点介绍了机器博弈的时间自适应分配策略。

1 时间分配的数学模型

目前,中国象棋的国内外赛事中,常常采用包干计时的方式。例如,每方基本时间 60 min,每走一步加时 30 s^[2]。

文中将以包干计时为例,说明时间的数学模型,以便更准确地刻画时间分配。

1) 红方第 k 步后的总时间

$$T_{\text{total}}^{\text{red}}(k) = T_0 + \cdot k.$$

式中: T_0 为各方基本时间, \cdot 为每步加时。

2) 红方前 k 步的总计用时

$$T_{\text{used}}^{\text{red}}(k) = \sum_{i=1}^k u(i).$$

式中: $u(i)$ 表示第 i 步着法用时, $i = 0, 1, \dots, k$ 。

3) 如果用 $T_{\text{used}}^{\text{red}}$ 表示红方本局总计用时 (K_F 为总步数),那么还有

$$T_{\text{used}}^{\text{red}}(K_F) = T_{\text{open}}^{\text{red}} + T_{\text{mid}}^{\text{red}} + T_{\text{end}}^{\text{red}}.$$

收稿日期:2006-07-22.

式中:下标 open, mid, end 分别代表了红方在开局、中局和残局阶段的耗时.

$$T_{open}^{red} = open \cdot K_{open},$$

$$T_{mid}^{red} = mid \cdot K_{mid},$$

$$T_{end}^{red} = end \cdot K_{end}.$$

式中: 为该阶段的每步平均耗时, K_{open} , K_{end} 则为开局和残局阶段的按谱走棋的步数. 在有开局库和残局库的情况下, 查库的时间消耗几乎可以忽略不计 (常常按表的时间要比查库的时间长得多). 可见, 开发内容丰富的开局库与残局库的优越性. 按库中棋谱中给出的着法走得越多, 亦即开局库越晚“脱谱”, 残局库越早“入谱”越有利. 因为不仅很少耗时, 而且还可以赢得更多的“每步加时”.

4) 红方第 k 步后的剩余时间

$$T_{left}^{red}(k) = T_{total}^{red}(k) - T_{used}^{red}(k-1) - u(k).$$

式中: $T_{used}^{red}(k-1)$ 为红方 k-1 步总计用时, $u(k)$ 为红方第 k 步用时. 应该说这是最被关心的.

2 时间的静态分配策略

时间的静态分配策略是根据事前给出的常量 (博弈树搜索的深度、已经访问过的局面数、所耗时间), 来决定何时停止搜索的策略.

2.1 搜索时间、搜索节点数、搜索深度之间的关系

为分析搜索时间、搜索节点数与博弈树搜索深度之间关系, 选 3 个具有代表性的局面 (开局、较复杂的局面、较简单的残局局面) 进行了测试, 得出关系曲线如图 1~3 所示.

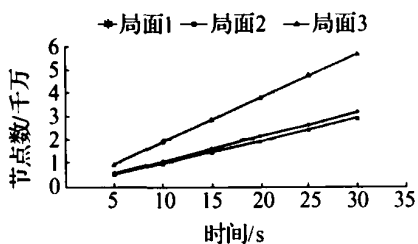


图1 时间与节点数关系图

Fig. 1 Relationships between used-time and the number of nodes

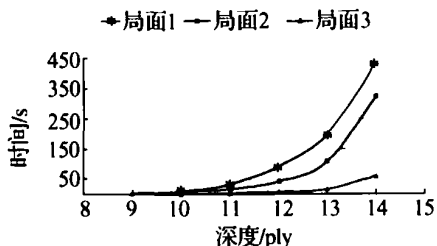


图2 博弈树深度与时间关系图

Fig. 2 Relationships between used-time and tree's depth searched

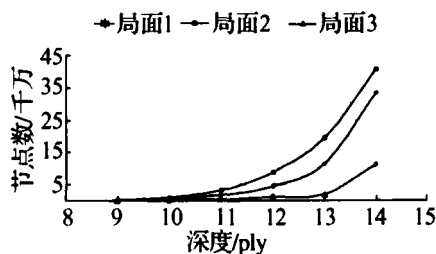


图3 博弈树深度与节点数关系图

Fig. 3 Relationships between depth of tree and the number of nodes searched

从图中不难看出:

- 1) 同一局面下时间与节点数大致成线性关系. 与残局相比, 初始局面和中局局面单位时间所搜索的节点数要少, 这是因为后者局面复杂, 评估函数开销较大.
- 2) 伴随着深度的增加, 博弈树搜索所花费的时间呈指数增长. 且尤以中局花费的时间的增长速度最快, 残局的增长速度最慢.
- 3) 伴随着深度的增加, 要搜索的博弈树节点的数目呈指数增长. 以中局节点数目增长最快, 残局增长速度最慢.

2.2 静态分配策略

- 1) 固定时限制: 比较符合人类棋手的习惯. 缺点是在中局搜索复杂局面的层数太浅, 而简化的残局常常可以搜得非常深.
- 2) 固定层数: 复杂局面每出一步棋, 要花费较多的时间搜索; 而简单局面, 如残局则很快出步.
- 3) 固定节点数: 不容易估计, 复杂局面搜索层数浅; 简单局面搜索得过深.

静态分配策略的优点在于简单, 易于实现. 缺点是不灵活 (需要事前固定搜索时间的长度, 或搜索深度, 或节点数)、分配不公 (无法兼顾简单局面与复杂局面)、无法预测是否会超时或者有大量时间剩余. 上述方法应用于比赛显得太过粗糙, 但在象棋程序的开发与调试阶段, 还是非常有用的.

3 时间的动态分配和调整策略

时间的静态分配策略缺乏自适应性, 而动态分配和调整策略则根据其策略目标, 为每一步着法分配不同的搜索时间. 其中, 极为简单、有效的办法是“平均配时”模型.

3.1 “平均配时”模型分析

作为棋手或机器, 下棋过程中必然时刻面对 2 个问题:

- 1) 下一阶段时间分配: 估计还有多少步, 大致时间

分配.

2) 本步可以花费多少时间.

显然这是没有统一答案的,只能根据具体棋类特点和经验进行处理.

对于时间控制的“平均配时”模型,常常就是估计一个系数 N ,则本步的用时时间便为

$$u(k) = T_{\text{left}}^{\text{red}}(k - 1) / N.$$

按照当前机器博弈国际赛事的规则,没有每步加时,则不难写出剩余时间的演化方程:

$$T_{\text{left}}^{\text{red}}(k) = T_{\text{left}}^{\text{red}}(k - 1) - T_{\text{left}}^{\text{red}}(k - 1) / N = (1 - 1 / N) T_{\text{left}}^{\text{red}}(k - 1).$$

这是一个递减等比级数.如果假设每方 30 min 包干,则在图 1 中给出了 N 为不同值时各步计算时间的变化曲线.

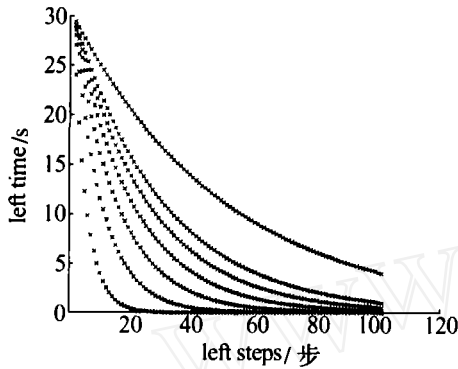


图 4 不同 N 值下步数与剩余时间的关系曲线图

Fig. 4 The functions of time left w. r. t moves for different N

分析图 4 中的曲线不难看出:

1) 由于曲线最终趋近于零,但永远不会到达零,理想状况(无延迟,无误差)按此方式设定时间是永远不会超时的.所以,具有一定的弹性;

2) 各步分时是“前松后紧”, N 值越小“前松后紧”的程度越严重;

3) 由于基本是按“平均”的思想分配时间,因此弹性不足,即缺少对于局面的自适应性;

4) 实际用时比起设定值而言是“只少不多”,所以此种分时方法偏于保守.

3.2 自适应时间分配策略

时间的自适应分配策略由 2 部分组成:预估计分配和自适应调整.

3.2.1 预估计分配

根据总时间和所用去的时间对本步搜索的用时进行分配,也就是给出一个期望搜索时间 E_t . 可以通过“平均配时”模型计算 E_t . 只要 N 值选取得当,就可以基本保证不会超时.“平均配时”模型还仅仅

从时间的数量上考虑,未涉及着法质量等因素.国际象棋 Crafty 和 Fruit 都采用“平均配时”模型计算 E_t 的大小.

3.2.2 搜索技术对时间分配的影响

迭代深化^[3-5]技术的发现,对于带有时间限制的博弈树搜索有很大帮助.

迭代深化的过程:先搜索深度为 2 层的博弈树(第 1 次迭代),然后搜索深度为 3 层的博弈树(第 2 次迭代),接着 4 层(第 3 次迭代),...,直到所分配的时间耗尽为止.时间用完时,程序会选择最后一次迭代的结果.

迭代深化的额外好处在于它方便着法排序.因此,迭代深化一般都和历史启发等着法排序算法结合运用.程序知道在前一次迭代之中哪个着法是最好的,在每一次新的迭代中都优先搜索主要变例(principal variation).前期迭代的额外时间开销不但没有浪费,反而会有更高的回报.这是因为着法按照从好到坏的顺序排列了,使博弈树由于高剪枝率而更接近于最小树.

对于时间控制问题,采用迭代深化技术还可以根据 2 次相邻迭代评估值的变化,及时判断时间需求的变化,以便调整时间分配.

此外,配合迭代深化采用着法排序对于精细的时间控制也大有裨益.着法按照从好到差的顺序基本有序,所搜索的迭代深度为 d 的第一个子根节点在很大程度上可以对应该深度搜索的最佳着法.也就是说,如果第一个子根节点着法搜索完毕且令人满意,且分配的时间用完了,就可以退出搜索,而不必完成全部子根节点着法的搜索.

3.2.3 自适应时间调整策略

机器下棋的时候,其棋力往往取决于搜索的深度和评估的准确性^[6-7].若评估是比较准确的,花在一歩棋上的时间越多,所得着法的质量就越好.众所周知,一根链条的张力取决于它最薄弱的那一节,全局时间是有限的,如何将时间恰当地分配给各个着法,使每个着法都有相对“均匀”的质量,从而达到全局最优,则是时间分配策略的最终目的.

时间的预估计分配,显然还只是局限于找到一个初步具有自适应性的事前分配的策略.统一的数学公式计算各步着法的时间并不符合各个着法的需要,只能发挥避免超时的作用.有时 E_t 用完,所得的着法质量不能令人放心;而有时又不需要耗时 E_t 就能得到相当满意的着法.为此,需要随时调节 E_t 的大小.增设一个新变量 M_t ,一旦用时超过了 M_t ,搜

索就要无条件停止. 显然,

$$E_{t_{red}} < M_{t_{red}} < T_{total}^{red}(k) - T_{used}^{red}(k-1), \\ u(k) < M_{t_{red}}.$$

M_t 通常都是 E_t 的若干倍. 那么 $(0, M_t)$ 便是 E_t 的自适应调节范围.

一般只有在第一个子根节点着法是最佳着法, 且在多次迭代中都证明其足够好(如, 没有 Fail-low 过)的时候, 才不需调整 E_t 的大小.

在采用迭代深化的搜索算法的程序中, 设搜索深度为 d 时, 得到的当前最佳着法的评估值为 Eva . 而迭代深度为 $d-1$ 的最佳着法评估值为 $Last_eva$. 则 $Last_eva - Eva$ 反映了由 $d-1$ 层迭代到 d 层的过程中, 评估值变化的剧烈程度. 为区别预估时分配的 E_t , 用新变量 T 表示调整之后的搜索时间, 初始时有 $T = E_t$. 那么, 用分段函数表示调整之后的时间

$$T = \begin{cases} 2 E_t, & 0 < Last_eva - Eva < 30 \\ 4 E_t, & 30 \leq Last_eva - Eva < 70 \\ M_t, & 70 \leq Last_eva - Eva \end{cases}$$

注: 分段函数中的评估值参照车(静态子力值 1 000)、马(静态子力值 400)、炮(静态子力值 440)、兵(静态子力值 60)取值.

从以上函数还可以看出, 由于时间的预估时分配策略趋于保守, 所以局面变差的时候, 自适应调整策略可以视情况而成倍地增加搜索时间. 与之相反, 减少搜索用时应该是很谨慎的. 一般只在没有风险或风险很小时, 才减少搜索用时. 下面列举几种这样的情况:

1) 换子. 对方子吃掉本方有根子, 变为无根子的时候, 绝大多数情况都要吃掉对方子, 几乎不需要多加考虑. 但为了安全起见, 可以稍用点时间继续推演几步. 如, 令搜索时间 $T = E_t/4$.

2) 本方只有一个着法可选, 也同样不需过多思考, 因为搜得再深也是徒劳. 如, 令搜索时间 $T = E_t/10$.

3) 循环探测模块探测出了循环着法(长将、长捉等), 不再搜索, 立刻出着.

4) 能在未来连续几招之内将对方将死的着法称为杀棋. 找到杀棋或自己被对方杀, 不再继续搜索, 立即出着.

在进行自适应调整策略的时候, 要先判断是不是遇到了特殊情形. 如果是, 先按特殊情形调整, 因为它们往往可以提前结束搜索; 否则, 按照分段函数进行调整.

此外, 调整还可以依据其他的条件. 比如, 尽管连续的 2 次迭代之间的估值变差程度较大, 但由于本方原来的优势大, 变化之后还是优于对手, 那么可以将时间延长的幅度适当缩小. 总之, 应该根据程序执行过程中一些指标的变化灵活调整.

自适应的调整策略严重依赖于评估. 如果评估不准确, 或者由于评估模块的大量改动, 可能会引起搜索用时的非正常波动. 在对机器博弈软件的调试过程中, 曾发现由于评估的改动, 导致一些平缓局面在迭代深化的搜索过程中的 $Last_eva - Eva$ 变化剧烈, 从而使每步用时大量增加, 最终超时, 这是需要注意的.

4 时间控制策略的实现

为了进行时间控制, 需要按前面讲述的过程进行检查. 若每搜索一个局面就检查一次, 开销将非常巨大, 完全不符合象棋程序设计精打细算的特点. 通常采用一个固定的节点数作为检测的单位, 比如平均搜索速度约为 100 000 节点/s 的引擎, 依据精度要求设置每隔 1 000、5 000、10 000 个节点检查一次.

还应注意, 时间分配与调整问题一方面与搜索和评估紧密相关, 另一方面还与个人经验和喜好紧密相关, 很难说哪个具体的时间分配和调整的方案是最佳的. 更确切地说, 不同方案展现出不同的下棋风格.

5 其他优化措施

在对手思考的时候, 机器也可以猜测对方的应着, 并按照对方应着继续思索. 如果猜测正确, 那么就极大地节省了时间, 这种方法称为后台思考(Ponder). 它相当于把对手的时间挪为己用, 实践证明是非常有效的.

丰富开局库, 开局晚脱谱, 也是节省时间的办法; 此外, 中国象棋残局库尚不普及, 目前大多数已有的残局库或多或少都存在不尽人意之处. 开发出实用的中国象棋残局库, 在残局时利用残局库也有同样的作用. 使用数据库实质是将在在线计算转化为离线计算, 变相地延伸了比赛时间.

6 结束语

总结了合理分配和利用时间的方法, 并建立了时间控制模型. 实践表明, 对于时间模型和配时策略应该给予足够的重视, 这是在有限资源的条件下提

高棋力的一个有效措施.此外,时间分配和调整也是对弈风格的一种体现.

参考文献:

[1]黄 晨.解剖大象的眼睛——中国象棋程序设计探索 [EB/OL]. http://www.elephantbase.net/computer/ele-eye_ponder.htm. 2005.11.

[2]王孔兴.象棋竞赛裁判手册[M].北京:人民体育出版社, 2001.

[3]SCHAEFFER J. The history heuristic and alpha-beta search enhancements in practice[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1989, 11 (11) : 1203 - 1212.

[4]REINEFELD A, MARSLAND T A. Enhanced iterative-deepening search[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1994,7(7) :701 - 710.

[5]KORF R E, REID M, EDEL KAMP S. Time complexity of iterative-deepening-A[J]. Artificial Intelligence, 2001, 129(1 - 2) :199 - 218.

[6]徐心和,王 骄.中国象棋计算机博弈关键技术分析[J].小型微型计算机系统,2006,27(6) :961 - 969.
XU Xinhe, WANG Jiao. Key technologies analysis of chinese chess computer game [J]. Mini-Micro Systems. 2006,27(6) :961 - 969.

[7]王小春. PC 游戏编程 [M]. 重庆:重庆大学出版社, 2002.

作者简介:



徐长明,男,1978年生,博士研究生,主要从事中国象棋的机器博弈、计算机网络等的研究。E-mail: changmingxu@gmail.com



南晓斐,女,1983年生,硕士研究生,主要从事机器博弈的研究。



王 骄,男,1978年生,博士研究生,研究方向为人工智能与机器博弈。



徐心和,男,1940年生,教授,博士生导师,研究方向为控制理论与应用、系统仿真、智能机器人、机器博弈等。

计算机应用研究

月刊

每月10日出版

邮发代号
62-68

中国科技核心期刊
 全国中文核心期刊
 中国科技论文统计源期刊
 中国计算机学会会刊
 中国期刊方阵双效期刊
 中国科学引文数据库来源期刊
 第二届国家期刊奖百种重点科技期刊

2007年扩容

高档纸张

精美印制

大16开本
 内文320页
 订价每月30元
 全年360元

欢迎赐稿 欢迎订阅

地址:成都市成科西路3号 邮编:610041 电话:028-85249567 网址:www.arocmag.com