

基于知识推理的博弈树搜索算法

蒋加伏 陈蔼祥 唐贤英

(长沙交通学院计算机工程系,长沙 410076)

摘要 针对搜索技术中存在的由于搜索空间过于巨大而引起的搜索效率下降的缺点,该文提出了一种把规则提取、知识推理、搜索相结合的新算法。并把该算法运用到中国象棋中去。实验结果表明,该算法不但能保证博弈水平,还能提高搜索效率。

关键词 博弈树 搜索算法 规则提取 知识推理

文章编号 1002-8331-(2004)01-0074-03 文献标识码 A 中图分类号 TP181

Search Algorithm for Game of Checkers Based on Knowledge Inference

Jiang Jiafu Chen Aixiang Tang Xianying

(Dept. of Computer Engineering, Changsha Jiaotong University, Changsha 410076)

Abstract: Search technology is an important part of Artificial Intelligence. The search efficiency will descend when the problem state is large. In this paper a new algorithm which combines the rule distill, Knowledge Inference and search is presented. we apply the algorithm to china chess, the result shows that the algorithm can not only guarantee the computer's game level but also improves its search efficiency.

Keywords: game of checkers search algorithm rule extract knowledge inference

1 引言

搜索算法是人工智能中的一个基本问题,是推理不可分割的一部分。启发式搜索是在搜索中加入了与问题有关的启发式信息,用于指导搜索朝着最有希望的方向前进,加速问题的求解过程并找到最优解。文献[1]中 J. Pearl 对启发式评价函数的估计精度与 A^* 的平均计算量之间的关系进行了全面的研究,结果表明使 A^* 的计算量为多项式的充要条件是:评价函数必须具有“对数”阶的精度。实际上,这个要求很难达到。这就是说, A^* 算法不能克服计算量的指数爆炸的困难。于是,文献[2]又在 Peral 的搜索概率模型的基础上,提出了一种叫统计启发式的搜索技术,即 SA 算法。SA 算法的基本思想是利用启发式搜索与统计推断的相似性,将搜索过程看作是一个随机取样的过程,将搜索过程中每取定一个搜索方向看成是进行一次统计推断,从而把统计推断中较成熟的技术移植到启发式搜索中来。

在现实中,常有这样一类问题,参加搜索的不只有一个主体,而是对抗性的敌我双方,这样,搜索的进程不仅取决于参加一方的“如意算盘”,而且取决于双方的应付策略,由此产生的搜索树,通常称为博弈树。通常使用的博弈树搜索算法有极大极小算法和 α - β 剪枝技术。但是由于博弈树搜索是一项复杂性极高的工作,大量的研究表明,极大极小值搜索算法和 α - β 剪枝技术同样存在许多缺点,文献[3]列出了诸如 α - β 剪枝技术可能在特定搜索中因子树排列次序不同而失效、估值函数误差向上传播会引起上级结点作决策时产生关键性的失误、无法反映弈手的作战意图以及由于搜索深度有限而引起不合理的结论等 7 大缺点。

以上各种搜索算法中,无论是博弈树搜索还是一般的搜索,其着眼点都仅是对当前格局(棋局)而进行考虑的,都只是运用了计算机计算速度快的特点来求解问题的。因此,一旦碰上问题解空间相当庞大,搜索范围相当广泛的时候,以上所有的搜索算法统统都不可避免地陷入计算量过于巨大这一缺点。事实上人类在下棋、博弈过程中,不但要充分运用自己的计算棋步的能力,而且更多的是带有经验性的智力行为。通常人在下棋时每走一步棋前,脑海里都有一种对当前棋局势的估计,再结合以前的下棋经验,综合运用自己的计算和推理能力,从而最终得出该不该走某步棋的判断。这就是说,人类在下棋过程中不仅考虑当前棋局,而且还充分利用自己的“记忆”功能,综合考虑以前的棋局来进行判断的。这就是为什么人类往往能战胜计算速度远快于自己的计算机的缘故。因此,前述的各种搜索算法都没有充分利用计算机所具有的“记忆”功能这一特点。基于此,该文提出了一种在搜索过程当中结合历史经验知识,运用计算机的推理技术进行推理的新算法,从而在一定程度上减少计算量和加快推理速度。

2 基于 Apriori 算法的博弈规则提取

博弈树就是指用来描述博弈过程的与/或树,它有以下特点:

(1) 初始节点对应于初始结局。

(2) 或“节点和”与“节点逐层交替出现,且轮流地扩展节点。

博弈树的搜索过程称为“极大极小过程”,其过程描述如下:

(1) 生成一定深度的博弈树。

基金项目 湖南省自然科学基金(编号 00JJY2059)资助

作者简介 蒋加伏(1964-),男,副教授,主要研究方向:分布式计算、人工智能、信号与图像处理。唐贤瑛(1941-),男,教授,主要研究方向:模式识别与人工智能。

(C)1994-2012 China Academic Journal Electronic Publishing House. All rights reserved. <http://www.cnki.net>

(2)对格局给出一个估价函数,并对端结点进行估值(静态估值)。

(3)计算上层结点的倒推值。对“或”结点,取其子结点的最大值(最有利于自己);对“与”结点,取其子结点的最小值(立足于最坏情况)。

(4)倒推值较大的方案,为我方当前最好的方案。

在博弈搜索算法中,常用的是 α - β 剪枝搜索,既在搜索过程中,通过边生成节点,边计算倒推值,从而可删去一些不必要的结点,达到提高搜索效率的目的。

虽然 α - β 剪枝技术在一定程度上加快了搜索过程,提高了搜索效率。但是,正如文献[3]所述,单一的博弈搜索算法有其固有的缺点,比如在一定条件下 α - β 剪枝会失效、无法反映弈手的作战意图等,再有就是单纯的博弈搜索算法不能很好地模拟人类进行下棋这一类智力博弈行为时的思维过程。为了模拟人类进行智力博弈时的思维过程,该文提出了基于Apriori算法的博弈规则提取算法。

2.1 Apriori 算法的相关概念

令 $I=\{i_1, i_2, \dots, i_n\}$ 为一个物品集, D 为一个事物数据库,其中每一个事物 $T \subset I$ 为一个物品集,每个事物用一个标识符TID表示:

(1)设 A, B 为物品集, $A \subset I, B \subset I$,且 $A \cap B = \emptyset$,则称蕴含关系 $A \Rightarrow B$ 为关联规则。

(2)设 S 为 D 中包含 $A \cup B$ 的事物的百分比,则称规则 $A \Rightarrow B$ 在事物数据库 D 中成立且有支持度 S ,且取为概率 $P(A \cup B)$,即 $\text{Support}(A \Rightarrow B) = P(A \cup B)$ 。

(3)设 C 为 D 中包含 A 也包含 B 的事物的百分比,则称规则 $A \Rightarrow B$ 在事物数据库 D 中具有可信度 C ,且取为概率 $P(B/A)$,即 $\text{Confience}(A \Rightarrow B) = P(B/A)$ 。

(4)含有 K 个物品的物品集,称为 K -物品集,如果一个物品集满足最小支持度,则称该物品集为常用物品集。常用 K -物品集用 L_k 表示。

2.2 关联规则提取

2.2.1 利用生成候选物品集求常用物品集

Apriori 性质

常用物品集的非空子集必定也是常用物品集。如果物品集 I 不是常用物品集,则添加物品 A 所得到的物品集 $I \cup A$ 也不是常用物品集。

Apriori 算法

结合:由 L_{k-1} 与自身结合,产生候选 K -物品集 C_k 。

删除:根据Apriori性质,将 C_k 中支持度小于最小支持度 min-sup 的物品集删去,由支持度计数不小于 min-sup 的物品集构成 L_k 。

2.2.2 从常用物品集提取关联规则

对于每个常用物品集 L ,生成 L 的所有非空子集;

对于每个 L 的非空子集 S ,如果:

$$\frac{\text{Support-Count}(C)}{\text{Support-Count}(S)} > \text{min-conf}$$

则输出规则 $S \Rightarrow (L-S)$

其中 min-conf 为最小可信度阈值。

2.3 博弈规则提取

(1)记录对弈双方所走的每一步棋。

(2)把每一盘棋看作是一个物品集,把同一盘棋的每一步

棋看作是一个物品,运用Apriori算法产生常用物品集。

(3)从常用物品集提取棋步的关联规则。

3 基于知识推理的博弈树搜索

在得到有关棋步的关联规则后,博弈过程中计算机就可以结合这些关联规则进行搜索。这些关联规则类似人的经验值,下的棋越多,经验值就越高。而经验值越高,意味着规则库中可用规则就越多。不仅如此,这些关联规则还可以体现弈手下棋时的作战意图,这一点在单纯的博弈搜索算法中是没法体现的。因此,可以得到基于知识推理的博弈搜索算法描述如下:

根据对手所走的棋步,看在规则库中是否有可用规则。若有一条或一条以上的规则可用,则随机选取一条规则直接执行。若无可用规则,则执行 α - β 剪枝搜索。

具体算法描述:

(1)搜索过程中记下每一盘棋对弈双方下棋走的棋步。

(2)每下完一盘棋后,把每一盘棋看作一个物品集,同一盘棋内的每一步棋看作一个物品,对所记录的棋局用Apriori算法进行关联规则提取,并把规则存入规则库中。

(3)博弈过程中,根据对手所走的棋步,先查找规则库中是否有可用规则。若有,则执行规则。否则,执行 α - β 剪枝搜索。

4 算法在中国象棋中的应用

该文在VC++平台上编程实现了上述算法在中国象棋中的应用。编程过程当中通过采用数据库技术,建立适当的数据库,把下棋过程当中所走的棋步保存到数据库中,然后在这些棋步数据库中提取关联规则,把得到的关联规则存进规则库中。这样,当规则库建立好以后,下棋过程当中,只要满足了规则库中的某条规则,就可以直接执行规则,不再计算。如果没有可用规则,则按一般情况进行搜索计算。

4.1 知识库的设计

该文设计了如下形式的知识库。首先,建立一个名为Knowledge的数据库,该数据库当中有若干个表,每个表对应着一盘棋。表中保存的是一盘棋当中对弈双方所走过的棋步。表中的字段有棋步、走子、吃子、起点横标、起点纵标、目标点横标、目标点纵标和处理标志位8个字段。表1是本实验中所选取出来的一个表。表中每个字段意义如下:

棋步:代表该步棋在这盘棋当中是第几步棋。

走子:代表所走的是那一个棋子的序号,中国象棋当中有32个棋,故用0~31这32个数字来代替。比如“0”这个数字代表象棋的“红帅”。

吃子:意义同上,不过用数字“32”代表这步棋的目标点上没有棋子存在。

起点横标、起点纵标:分别代表走棋时的起点横坐标和纵坐标。

目标点横标、目标点纵标:分别代表走棋时的目标点横坐标和纵坐标。

处理标志位:代表该表是否已经通过了相应的处理,该标志位若为“0”则表示该表还没有经过处理,若为非“0”,则表示已经经过处理,且其数值代表了该步棋在后面将要说明的一个为workspace的数据库当中所对应的棋步序号。比如表1中棋步“1”的处理标志位为“90”,这表示该表已经经过处理,而且该表中的第一步棋在workspace数据库中对应的是第90步棋。

表 1 Knowledge 数据库中的一个表

棋步	走子	吃子	起点横标	起点纵标	目标点横标	目标点纵标	处理标志位	棋步	走子	吃子	起点横标	起点纵标	目标点横标	目标点纵标	处理标志位
1	10	32	8	8	5	8	90	32	25	32	2	5	2	4	458
2	21	32	2	1	3	3	26	33	8	32	7	4	7	5	459
3	5	32	2	10	3	8	92	34	25	32	2	4	2	8	460
4	24	32	9	1	9	2	208	35	7	32	1	2	2	2	461
5	12	32	3	7	3	6	91	36	25	32	2	8	4	8	462
6	25	32	2	3	2	5	283	37	7	32	2	2	4	2	463
7	6	32	8	10	7	8	94	38	26	32	5	3	4	3	464
8	26	32	8	3	5	3	442	39	7	32	4	2	3	2	465
9	8	32	9	10	8	10	11	40	19	32	3	1	5	3	106
10	22	32	8	1	7	3	2	41	23	6	6	8	7	8	466
11	8	32	8	10	8	4	100	42	5	23	5	9	7	8	467
12	24	32	9	2	9	3	443	43	8	32	7	5	2	5	468
13	8	30	8	4	7	4	103	44	23	6	6	8	7	8	466
14	23	32	1	1	1	2	282	45	5	23	5	9	7	8	467
15	7	32	1	10	1	9	33	46	25	5	4	8	7	8	469
16	23	32	1	2	6	2	444	47	8	32	2	5	2	1	470
17	7	32	1	9	4	9	445	48	26	32	4	3	4	1	471
18	17	32	4	1	5	2	259	49	8	32	2	1	2	4	472
19	7	32	4	9	4	4	446	50	31	32	9	4	9	5	72
20	23	32	6	2	6	8	447	51	8	32	2	4	4	4	473
21	5	32	3	8	5	9	448	52	22	32	7	3	6	5	274
22	23	32	6	8	6	9	449	53	9	32	2	9	2	1	474
23	9	32	2	8	2	9	450	54	26	1	4	1	4	10	475
24	23	32	6	9	6	8	451	55	7	32	3	2	3	1	476
25	7	28	4	4	3	4	452	56	17	32	5	2	4	1	267
26	21	32	3	3	1	2	453	57	8	17	4	4	4	1	477
27	7	32	3	4	3	2	454	58	16	32	5	1	5	2	171
28	25	32	2	5	1	5	455	59	7	32	3	1	3	2	478
29	11	32	1	7	1	6	192	60	26	32	4	10	4	2	479
30	25	32	1	5	2	5	456	61	8	26	4	1	4	2	480
31	7	21	3	2	1	2	457	62	19	32	5	3	7	5	121

4.2 workspace 数据库的设计

为了便于用 Apriori 算法进行规则提取，需要给每一步棋取个唯一的“名字”。Workspace 数据库中的 steptname 表实现了棋步到棋步序号的一一对应关系。这样，通过这个表，只要给定一个棋步的序号，就可以映射到该棋步序号所对应的棋步。其中表 steptname 中有物品标识、走子、吃子、起点横标、起点纵标、目标点横标、目标点纵标 7 个字段。其中“物品标识”字段即对应着棋步的序号，其余字段同前所述。见表 2。

表 2 Workspace 数据库中的 steptname 表

物品标识	走子	吃子	起点横标	起点纵标	目标点横标	目标点纵标
------	----	----	------	------	-------	-------

4.3 rule 数据库的设计

通过规则提取后所得到的规则同样是存在一个名为 rule 的数据库中。该数据库有一个名为 ruletable 的表。表中的字段带 IF 的是规则的前件，带 THEN 的是规则的结论。IF 字段和 THEN 字段的数字代表的是某步棋在 workspace 数据库中的棋步序号（即 steptname 表中的物品标识字段值）。表 3 是通过在 8 盘棋的数据中所提取出来的一些关联规则。

5 实验结果分析及结论

从实验结果来看，程序能够提取出知识库中一些常用的关联规则，比如上面的规则表中的第 8 条规则是 IF“90”THEN“26”，翻译成棋谱的术语就是“如果炮二平五，那么马二进三”，这是下棋时经常使用的一着应对策略。并且，一旦在规则的前

表 3 运用 Apriori 算法所提取的关联规则

IF	IF 走子	IF 吃子	IF 起点横标	IF 起点纵标	IF 目标点横标	IF 目标点纵标	THEN	THEN 走子	THEN 吃子	THEN 起点横标	THEN 起点纵标	THEN 目标点横标	THEN 目标点纵标
11	8	32	9	10	8	10	26	21	32	2	1	3	3
11	8	32	9	10	8	10	72	31	32	9	4	9	5
26	21	32	2	1	3	3	11	8	32	9	10	8	10
26	21	32	2	1	3	3	90	10	32	8	8	5	8
26	21	32	2	1	3	3	91	12	32	3	7	3	6
26	21	32	2	1	3	3	92	5	32	2	10	3	8
72	31	32	9	4	9	5	11	8	32	9	10	8	10
90	10	32	8	8	5	8	26	21	32	2	1	3	3
91	12	32	3	7	3	6	26	21	32	2	1	3	3
92	5	32	2	10	3	8	26	21	32	2	1	3	3

对于绘图的数据信息,不仅包括图形的一些坐标信息,还包括图形的一些属性,如颜色、线宽、等。对于文本的数据信息,还包括字体、式样、大小等参数。

4 总结

在真实网络环境下,包括 100Mbps 以太网和校园网环境下,共享白板工具都获得了较为满意的预测结果。共享白板工具的共享数据传输具有较高的可靠性,同时系统也有灵活的发言权控制,远程指针有较好的处理。目前关于白板数据的导出处理还有待于完善,另外在网络传播时延控制方面也需要进一步研究。笔者正在作进一步工作争取系统的优化。

(上接 53 页)

- 1.冯培梯.系统辨识[M].杭州:浙江大学出版社,1999
- 2.Ljung L.System identification—Theory for the user[M].Second Edition, New Jersey Prentice Hall,1999
- 3.杨智民,王旭,庄显义.遗传算法在自动控制领域中的应用综述[J].信息与控制,2000,29(4):329~339
- 4.李茶玲,孙德保.遗传算法在系统辨识中的应用[J].华中理工大学学报,1998,26(7):57~58,70
- 5.姜波,汪秉文.基于遗传算法的非线性系统模型参数估计[J].控制理论与应用,2000,17(1):150~152
- 6.王小平,曹立明.遗传算法—理论、应用与软件实现[M].西安:西安交通大学出版社,2002
- 7.李敏强,寇纪淞,林丹等.遗传算法的基本理论与应用[M].北京:科学出版社,2002
- 8.Hatanaka T,Uosaki K,Tanaka H et al.System parameter estimation by evolutionary strategy[C].In Proc 35th SICE Annual Conf,Tottori, Japan,1996:1045~1048
- 9.Hatanaka T,Uosaki K,Koga M.Evolutionary computation approach to

(上接 76 页)

件得到满足之后,算法则直接执行规则的结论。这样,能够减少不必要的搜索,加快计算速度。当然,同样的炮二平五,可能有若干种不同的应对着数。这样,就会出现规则中前件相同,而结论不一样。对于这种规则,该文采取了随机操作的办法。就是说,如果有若干条规则的前件相同,则在执行规则时采取随机选取规则执行的办法。这也体现了人类下棋时在一定程度上的经验性和不确定性。

目前在启发式搜索算法中,其启发式估值都是基于当前棋局进行考虑的,这存在很大程度的不合理性。事实上,启发式估值当中应该包含有根据“经验”进行估计得到的一项估值。因此,把“经验”估值和启发式搜索算法结合起来是提高计算机博弈水平和加快计算机搜索速度的一条可行途径。该算法采取的是当走某步时先作经验判断,当经验判断失败时才采用一般的搜索算法进行搜索的办法,从而有效地把人工智能中的规则提取、推理、搜索技术有机地结合起来,实验结果证明该算法是有效的。事实上,更一般的做法应当是把“经验”估值也当作启发

(收稿日期:2002 年 12 月)

参考文献

- 1.David R Brow,Brad Vander Zanden.The Whiteboard Environment: An Electronic Sketchpad for Data Structure Design and Algorithm Description[C].In Proceedings 1998 IEEE Symposium on Visual Languages,1998:288~295
- 2.Bandoh H,Nemoto H,Sawada S et al.Advanced Learning Technologies[C].In Proceedings of International Workshop on IWALT,2000:41~44
- 3.Brent B Welch,Tcl/Tk 编程权威指南[M]
- 4.宏春,陈愚.多媒体会议系统共享白板功能的设计与实现[J].微型机及应用,1997(2):15~16,25

- Wiener model identification[C].In Proc 2002 Congress on Evolutionary Computation,Piscataway,NJ:IEEE Press,2002:914~919
- 10.Yao X,Liu Y.Fast evolution strategies[J].Control and Cybernetics,1997,26(3):467~496
- 11.Yao X,Liu Y,Lin G M.Evolutionary programming made faster[J].IEEE Trans Evolutionary Computation,1999,3(2):82~102
- 12.Chellapilla K,Fogel D B eds.Two new mutation operators for enhanced search and optimization in evolutionary programming[C].In Bosacchi B,Bezdek J C,Fogel D B eds.SPIE Proc Applications of Soft Computing,Bellingham SPIE Press,1997
- 13.王凌,郑大钟.一类改进进化规划及其优化性能分析[J].计算机工程与应用,2002,38(1):8~10
- 14.贾立,俞金寿.基于自适应混合变异进化策略的神经模糊系统及应用研究[J].系统仿真学报,2001,13(增刊):122~125
- 15.鄢烈祥.列队竞争算法解组合优化问题[J].湖北工学院学报,2000,15(2):1~4
- 16.Hsia T C 著,吴礼民译.系统辨识与应用[M].长沙:中南工业大学出版社,1986

式信息当中的一部分,即在估值函数 $f(n)=g(n)+h(n)$ 中应该加上 $f(n)$ 项,则估值函数应为 $f(n)=g(n)+h(n)+f(n)$,其中 $f(n)$ 代表通过“经验”估值得到的启发式信息。此外,为提高“经验”判断(即所提取的规则的正确性),对棋步的考虑就不应当仅是“走子”、“吃子”、“起点坐标”、“目标点坐标”这几项因素,还应考虑棋的局面情况。因为即使同样的一步棋,在不同的局面情况下可能会有不同的效果,同时,不同的棋步可能会有相同的战略意图。因此只有结合棋的局面情况,才能正确判断两步棋是否在战略意义上的“等价”。为此,可以引进模糊推理和分类的办法加以改进。(收稿日期:2003 年 10 月)

参考文献

- 1.陆汝钊.人工智能[M].北京:科学出版社,1999:354~355
- 2.张钹,张玲.问题求解理论及应用[M].北京:清华大学出版社,2000:365
- 3.蔡自兴,徐光佑.人工智能及其应用[M].北京:清华大学出版社,1999:60~72
- 4.Pear J.Heuristics Intelligent Search Strategies for Computer Problem Solving[M].Addison-Wesley Publishing Company,1984