

- ◆ 兼顾基础，重在提高与技巧
- ◆ 解惑答疑，提升系统开发水平



C/C++

中国象棋程序 入门与提高

蒋 鹏 雷贻祥 陈园园 著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

代码的力量——

C/C++中国象棋程序

入门与提高

蒋 鹏 雷贻祥 陈园园 著

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书由浅入深地介绍了中国象棋博弈程序的各个基本知识点，以实际案例来促进读者对算法的理解，提高实际编程能力。主要内容包括：中国象棋博弈，局面表示，走法表示及生成走法，局面评估，基本搜索算法，人机博弈，机器对弈，置换表，算法分析及测试技术，时间控制策略，启发式搜索策略，更多搜索策略。

本书适用于在校计算机专业本科学生及研究生，以及程序设计、算法、博弈和人工智能的爱好者及专业人士。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目（CIP）数据

C/C++中国象棋程序入门与提高 / 蒋鹏等著. —北京：电子工业出版社，2009.5
(代码的力量)

ISBN 978-7-121-08575-8

I. C… II. 蒋… III. ①C 语言—程序设计②中国象棋—应用程序—程序设计 IV. TP312 G891.2-39

中国版本图书馆 CIP 数据核字（2009）第 044870 号

责任编辑：段春荣

印 刷：北京市天竺颖华印刷厂

装 订：三河市鑫金马印装有限公司

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本：787×1092 1/16 印张：21.5 字数：550 千字

印 次：2009 年 5 月第 1 次印刷

定 价：35.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：(010) 88258888。

前言

程序设计是计算机专业的必修课，从 C 到 C++，再到可视化程序设计 VC++。C/C++ 是大学计算机专业的基础语言课，不少程序员都是从 C 走上他们的程序设计之路的。大学课程主要从语法的角度讲解 C/C++。具体是 C 讲述基础的语言设计方法，C++ 主要讲述面向对象程序设计方法，VC++ 则讲述基本的可视化程序设计的知识。太多的语法基础占据了语言课的大部分时间，学生学习后基本只能应付课后习题，要提高程序设计能力还有很长的路要走。学生一路走来，到毕业时很多人竟不能独立编写一个像样的程序。

数据结构，算法分析与设计则是计算机专业的主要课程，可许多学生学习之后竟然觉得数据结构无用。理论教学与实践的脱离使得学生只追求盲目的 60 分。很多计算机专业学生毕业之后没有独立程序设计的能力，无法从事计算机相关行业。

案例教学成为近年比较流行的教学模式。它将课本理论与实际项目开发结合在一起，提高了学生的动手能力和实践经验。实践证明，编写一个带有实际意义的项目是提高编程能力最高效的方法。如果能有一个项目将 C、C++、VC++、数据结构、算法分析与设计结合在一起，提高学生的基础程序设计能力，同时又能掌握数据结构与算法的高级应用，必然为以后从事软件开发打下良好的基础。中国象棋程序就是这样一个项目，它综合了 C、C++、VC++、数据结构、算法分析与设计等知识。象棋作为竞技体育同时又作为一个游戏，具有一定的趣味性。学生通过自己亲身参与游戏程序的编写可以提高学习的积极性。

中国象棋程序主要由局面表示、走法表示及生成、局面评估、搜索算法、界面控制等五大部分组成。

局面就是一盘棋经过若干回合之后当前所处的形势，包括棋盘、红黑双方所剩棋子及其在棋盘上的分布、当前该走棋一方、双方所剩时间、双方所剩走棋步数等内容。局面表示是象棋程序的基础，局面表示的好坏直接关系到走法生成、局面评估和搜索算法的效率，从而影响象棋程序得到的最佳走法。

象棋程序每一次思考的目的是获取一个最佳走法（至少在程序看来是最佳的）。要实现这一目的的简单方法就是生成全部所有可能的走法，然后再一个一个的比较，找出最佳的一个。实际上程序也是这样做的。

局面评估就是判断局面对红方（或者黑方，或者是当该前走棋一方）的优势，并把优势进行量化。由于象棋程序搜索复杂度太大，搜索函数不可能搜索到棋局终了的状态，所以必须在某个深度的结点上结束并返回上一层。这个结点并没有达到棋局结束（胜平负），应该给它一个值，反映局面状况，对红方有利还是对黑方有利，有多少优势。必须把这种优势量化，以便不同结点的优势可以进行比较，以确定哪一个结点更好。

中国象棋的局面变化实在是太多了，有时候一个局面可能走法达 100 多种，一般局面也有 40 多种走法。要完全搜索 10 步棋需要 3.3 年，即使完全搜索 7 步棋也要 27 分钟（按每秒搜索 10^8 个结点计）。按一盘棋平均 100 步（50 个回合）计，要完全搜索 100 步是绝对不可能的。如何让计算机在有限的时间内搜索到更多的空间和更深的步数，是计算机博弈程序必须考虑的问题。这除了与计算机硬件有关之外，与搜索算法关系很大。这是因为在搜索树空间中有些分支是多余的，搜索的时候可以跳过某些分支。跳过的分支越多，搜索的速度越快，但漏掉最优解的可能性也在增加。搜索算法必须又快又准地找到最优解。

一般棋手不可能理解计算机内的数据结构，同样计算机也很难看懂普通的象棋棋盘，所以要实现人与计算机博弈程序的较量，必须要有一个中介在人与计算机博弈程序之间传递信息。这个中介就是一个图形界面程序。它将局面及计算机走棋以图形界面展示给棋手，将棋手的走棋以特定数据结构传递给计算机博弈程序。博弈程序可以自带一个图形界面，它接收用户（棋手）的输入，即棋手所走的棋，然后将自己思考后所走的棋以图形显示在屏幕上，展示给棋手。

笔者多年从事 C、C++、VC++、数据结构、算法分析与设计的教学工作，同时也有多年的软件开发经验。此书以提高 C/C++ 程序设计能力为主要目标，深入浅出地介绍中国象棋博弈中的基本算法，分析算法实现的关键技术，再逐步介绍各种高级技巧，使读者能够迅速领会象棋程序的特点，自己动手写出更高效的程序。

所有算法均配有示例程序，使读者能够由浅入深地掌握中国象棋程序的要领。将编程与下象棋的实践紧密结合，有较强的趣味性。读者如果具有一定的数据结构基础知识就更容易理解本书。

读者朋友如有任何疑问和建议，可以到如下网站的本书讨论区同笔者一起探讨：

<http://www.pubeta.com>

作 者

2009 年 3 月

目 录

C O N T E N T S

第 1 章 绪论.....	1
1.1 机器博弈.....	1
1.1.1 Tic-Tac-Toe 游戏.....	2
1.1.2 国际象棋机器博弈.....	4
1.1.3 机器博弈发展趋势.....	5
1.2 中国象棋程序.....	6
1.2.1 中国象棋博弈程序组成.....	6
1.2.2 中国象棋程序研究现状.....	6
1.2.3 全国计算机博弈锦标赛.....	7
1.3 C/C++ 基础知识.....	8
1.3.1 结构体.....	8
1.3.2 联合体.....	10
1.3.3 枚举.....	11
1.3.4 指针.....	11
1.3.5 面向对象程序设计.....	11
1.4 数据结构基础知识.....	12
1.4.1 线性表.....	13
1.4.2 栈和队列.....	14
1.4.3 树.....	14
1.4.4 查找.....	15
1.4.5 排序.....	15
1.5 算法分析基础知识.....	16
1.5.1 算法描述.....	16
1.5.2 算法时间复杂度分析.....	18
第 2 章 中国象棋博弈.....	21
2.1 中国象棋简介.....	21
2.1.1 棋盘与棋子.....	21
2.1.2 走棋和吃子.....	22

2.1.3	将死和困毙	22
2.1.4	胜、负、和	22
2.2	中国象棋博弈程序	23
2.2.1	局面表示	24
2.2.2	走法生成	25
2.2.3	搜索算法	25
2.2.4	局面评估	26
第 3 章	局面表示	29
3.1	简单的表示方法	29
3.2	扩展数组表示	31
3.2.1	棋盘表示	31
3.2.2	棋子表示	32
3.2.3	二维数组与一维数组	34
3.3	字符串表示局面	34
3.3.1	棋子表示	35
3.3.2	棋盘表示	35
3.3.3	走方表示	36
3.3.4	走棋步数	36
3.4	不同表示方法的转换	36
3.4.1	一维数组转换成 FEN 串	37
3.4.2	FEN 串转换成一维数组	40
第 4 章	走法表示及生成走法	45
4.1	走法表示	45
4.2	车炮马象（相）士（仕）卒（兵）将（帅）走法生成	46
4.2.1	马的走法生成	46
4.2.2	将（帅）的走法生成	49
4.2.3	士（仕）的走法生成	50
4.2.4	象（相）的走法生成	52
4.2.5	车的走法生成	54
4.2.6	炮的走法生成	56
4.2.7	卒（兵）的走法生成	58
4.3	产生一个局面的全部走法	61
4.4	简化合理位置数组	63
4.5	棋子数组	67
4.6	将军检测	72
4.7	如何更快地生成走法	79
4.7.1	事先生成法	79

4.7.2 位行位列	80
第 5 章 局面评估	83
5.1 简单的局面评估算法	83
5.2 带棋子数组的评估	86
5.3 新的价值数组	87
5.4 位置分值	88
5.5 灵活性分值	94
5.6 更为复杂的局面评估	98
5.7 知识与速度	99
第 6 章 基本搜索算法	101
6.1 搜索树	101
6.2 深度优先搜索与广度优先搜索	102
6.3 简单的两步搜索	104
6.4 极大点与极小点	104
6.5 结点的层次	106
6.6 极大极小搜索算法	106
6.7 局面变换	108
6.7.1 用局部变量来保存局面	108
6.7.2 用全局变量来保存局面	109
6.8 走法栈	110
6.9 获取最佳走法	111
6.10 完整的搜索过程	113
6.11 合并极大点与极小点搜索	124
6.12 负极大值搜索	125
6.13 极大极小搜索时间分析	129
6.14 搜索剪枝	133
6.15 Alpha-Beta 搜索	136
6.16 Alpha-Beta 搜索时间分析	138
6.17 alpha 一直小于 beta 吗?	140
第 7 章 人机博弈	141
7.1 基本知识	141
7.1.1 程序流程	141
7.1.2 棋局状态	142
7.1.3 图形界面开发工具	142
7.2 VC++ 工程	143
7.2.1 创建 VC++ 工程	143

7.2.2	VC++工程文件	145
7.3	棋盘显示	147
7.3.1	加载图片	147
7.3.2	棋盘显示	150
7.4	计算机走棋	154
7.4.1	添加类	154
7.4.2	添加类的成员	156
7.4.3	添加走法结构	158
7.5	走法显示	158
7.6	棋手走棋	160
7.7	时间处理	164
7.7.1	计时策略	164
7.7.2	WM_TIMER 消息	165
7.7.3	显示时间	166
第 8 章	机器对弈——博弈引擎	167
8.1	UCCI 协议	167
8.1.1	通信方式	167
8.1.2	引擎状态	168
8.1.3	指令	168
8.1.4	反馈	169
8.2	常用指令和反馈	169
8.2.1	position	169
8.2.2	banmoves	170
8.2.3	go	170
8.2.4	bestmove	170
8.3	管道	171
8.3.1	创建管道	172
8.3.2	读写管道	174
8.4	UCCI 棋盘表示	177
8.4.1	棋盘的坐标表示	177
8.4.2	走法转换	178
8.5	博弈引擎	179
8.5.1	引擎程序运行方式	179
8.5.2	通信处理	179
8.5.3	协议处理	183
8.5.4	工作流程	183
8.6	界面程序	186
8.6.1	功能描述	186

8.6.2	加载引擎	187
8.6.3	卸载引擎	189
8.6.4	常用功能	189
第 9 章	置换表	193
9.1	置换表	194
9.2	哈希表	194
9.2.1	存储	194
9.2.2	查找	195
9.2.3	冲突	195
9.2.4	冲突处理方法	195
9.2.5	影响哈希表效率的因素	196
9.3	Zobrist 键值	197
9.4	哈希函数	198
9.5	结合置换表的 Alpha-Beta 搜索	200
9.6	结点深度	205
9.7	Alpha 结点和 beta 结点	208
9.8	最佳走法	214
9.9	获胜局面	219
9.10	超出边界的 Alpha-Beta 搜索	225
9.11	哈希表的冲突处理策略	230
9.12	清空哈希表	232
第 10 章	算法分析及测试技术	233
10.1	测试内容	233
10.2	测试用例设计	234
10.2.1	开局	234
10.2.2	中局	235
10.2.3	残局	236
10.3	测试代码	237
10.3.1	负极大值搜索	237
10.3.2	Alpha-Beta 搜索	241
10.3.3	结合置换表的 Alpha-Beta 搜索	245
10.4	测试结果分析	253
10.4.1	死亡结点	253
10.4.2	置换表的不稳定性	256
10.4.3	三种算法对照分析	256

第 11 章 时间控制策略	259
11.1 带时限的搜索算法.....	259
11.2 平均时间分配	260
11.3 迭代深化(Iterative Deepening)	263
11.3.1 限定时间内究竟能搜索多深.....	263
11.3.2 迭代深化.....	265
11.3.3 迭代深化时间分析.....	267
11.4 动态时间分配	267
11.5 结合置换表的限时搜索.....	269
第 12 章 启发式搜索策略	273
12.1 杀手启发 (Killer Heuristic)	273
12.2 历史表启发 (History Heuristic)	274
12.3 走法排序	275
12.3.1 吃子走法和不吃子走法.....	275
12.3.2 新的走法数组	287
12.3.3 吃子走法价值	292
12.3.4 不吃子走法的价值	298
12.3.5 走法排序	299
12.4 克服水平线效应	301
12.5 空着	306
12.6 开局库	307
12.6.1 开局库文件	307
12.6.2 开局库在内存中的存储.....	308
12.6.3 读取开局库文件	308
12.6.4 获取开局库走法	312
12.6.5 修改相关函数	314
12.7 残局库	322
第 13 章 更多搜索策略	323
13.1 PVS 主要变例搜索	323
13.2 MTD(f)算法	326
13.3 后台思考	332
13.4 最小树	332
13.5 你的策略	333
13.6 博弈程序的智能水平	333
参考文献	334

第1章 绪 论

中国象棋，一个简单而古老的游戏，在中国流传久远。下棋时，红方先走，黑方再走，然后红黑双方轮流走棋。一方的将（帅）被吃掉，或者走棋后双方将帅直接照面，或者无棋可走困毙，即为输，对方赢。红方走棋的时候，为了赢棋，得仔细考虑黑方的想法，而黑方走棋时也得考虑红方的想法，所以红方还得想到黑方在想他的想法，黑方当然也知道红方想到了他在想红方的想法。高明的棋手会相互揣摩、相互牵制，最大化自己的利益同时最小化对方的利益，如此就下出诸多精彩纷呈、变化多端的棋局。就这样，双方斗智头勇，轮流走棋，称为博弈。

博弈论是研究棋手们走棋中理性化、逻辑化的部分，并将其系统化为一门科学。数学家们将具体的问题抽象化，建立完备的逻辑框架体系，研究其规律。经济学家将博弈理论应用到经济领域，建立起很多经典的理论，如纳什平衡，极大促进经济学的发展，使博弈论极大程度的得到应用。以致现在说到博弈论，人们更多的想到的是经济学领域的博弈。

博弈是伴随着棋类游戏而产生的，当计算机诞生时，人们就已经想到用计算机来下棋，这就是机器博弈。机器博弈就是计算机下棋，将棋局状态、行棋规则输入计算机，由计算机根据规则进行思考，然后再给出一步合理的棋步。

本章主要内容

- 机器博弈概述。
- 中国象棋程序现状。
- C/C++基础知识。
- 数据结构基础知识。
- 算法分析基础知识。

1.1 机 器 博 弈

机器博弈是博弈游戏与计算机技术的结合，是人工智能领域里一个重要研究领域，在国际上已经开展了半个多世纪。象棋是从两军对阵中抽象出来的一种智力游戏，因此它是博弈的一个标准问题。下棋的双方无时不在调动自己的一切智能，充分发挥逻辑思维、形象思维和灵感思维的能力。所以，在人工智能领域始终将棋类的机器博弈作为最具挑战性的研究方向之一。

计算机是出现在国外的，国际象棋在国际上开展较广泛，因此国际象棋机器博弈的研究已经有很长一段时间。中国象棋虽是世界上历史最为悠久的棋类，早在两千多年前的战国时代就已经有了关于象棋的记载，但是计算机的普及及计算机技术的发展远远落后于国外先进

国家。中国象棋计算机博弈近几年才起步，参与者少，缺少参考文献。仅有少数商家开发了游戏软件和教学载体。中国象棋的计算机博弈还十分稚嫩，水平还不太高，开展得更不普及，这便是当前我们所面临的艰难局面。

1.1.1 Tic-Tac-Toe 游戏

最简单的博弈游戏当属 Tic-Tac-Toe 游戏。在 3×3 的矩形棋盘中共九个位置，双方轮流布棋，某方三个棋子连成一条线（直线，对角线）为胜，均无三子连成直线为平。

给每一个位置编号，如图 1-1 所示。

1	2	3
4	5	6
7	8	9

图 1-1 Tic-Tac-Toe 游戏棋盘

下面给出一个游戏例子，红方走位置 1、2、7、9，黑方走位置 5、3、6、4。下至第 8 手，黑方在第 2 行连成一条线，获胜，如图 1-2 所示。

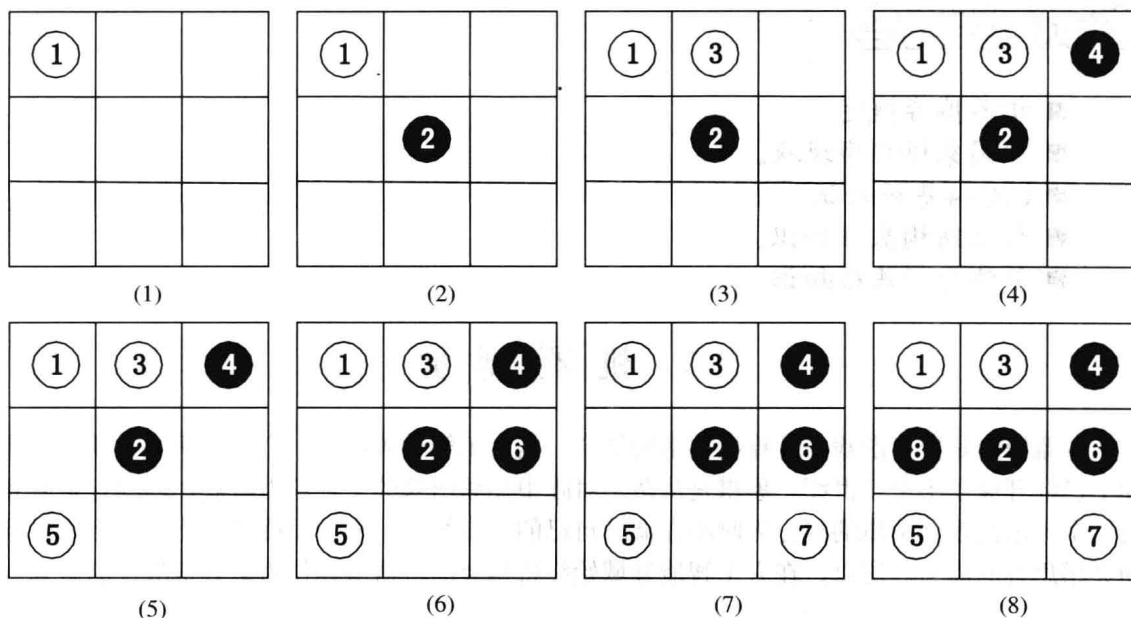


图 1-2 Tic-Tac-Toe 游戏走子顺序 1 示意图

再一个例子，红方走位置 1、6、3、9，黑方走位置 5、7、2。下至第 7 手，红方在第 3 列连成一条线，获胜，如图 1-3 所示。

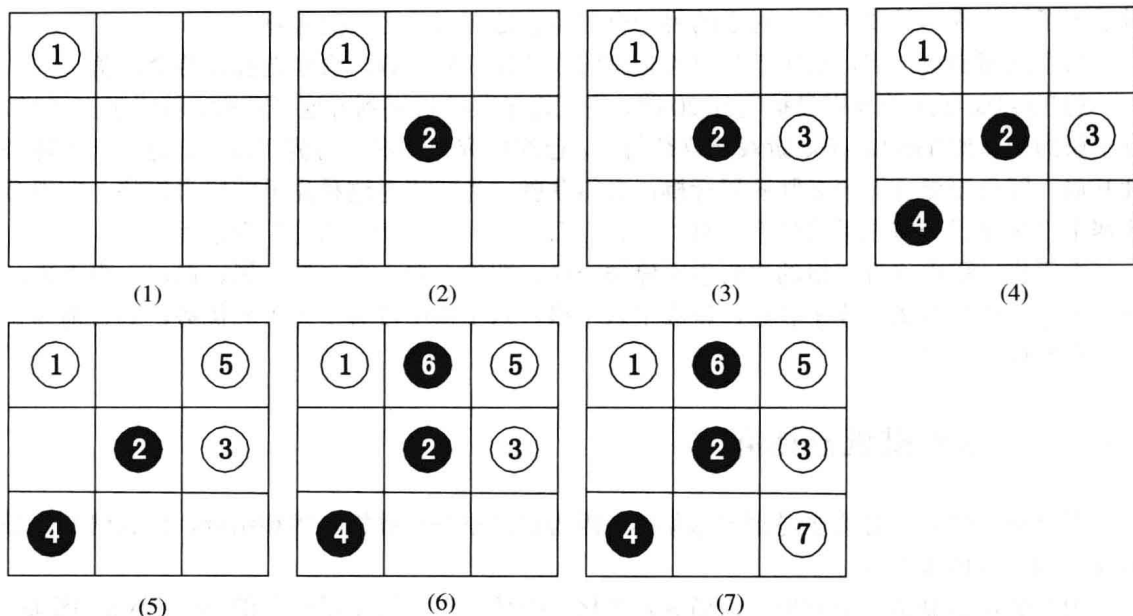


图 1-3 Tic-Tac-Toe 游戏走子顺序 2 示意图

时间复杂性分析

第 1 手：棋盘上无子，可以在任意 9 个位置落子。

第 2 手：棋盘有 1 子，可以在剩下的任意 8 个位置落子。

第 3 手：棋盘有 2 子，可以在剩下的任意 7 个位置落子。

第 4 手：棋盘有 3 子，可以在剩下的任意 6 个位置落子。

第 5 手：棋盘有 4 子，可以在剩下的任意 5 个位置落子。

第 6 手：棋盘有 5 子，可以在剩下的任意 4 个位置落子。

第 7 手：棋盘有 6 子，可以在剩下的任意 3 个位置落子。

第 8 手：棋盘有 7 子，可以在剩下的任意 2 个位置落子。

第 9 手：棋盘有 8 子，只能在剩下的 1 个位置落子。

第 1 子有 9 种可能位置，第 2 子有 8 种可能位置，第 3 子有 7 种可能位置，……第 8 子有两种可能位置，最后一子只有 1 个位置。所以最大的搜索空间为：

$$9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1 = 362880$$

以 CPU 每秒运算 1 亿次，现在的计算机一眨眼功夫就可以搜索完毕。已经知道 Tic-Tac-Toe 游戏没有必胜局面，因此计算机玩这种游戏可以说永不言败。即使人掌握了一些基本技巧也很难说败。

行棋技巧

先手玩家的最佳走法是，第 1 步最好下在四个角落（即 1、3、7、9），因为后手玩家除了下在中央的 5 号位置之外必败。即使对手下了该位置，只要回以马步布局或对角布局也还有一半的胜算。

先手玩家第 1 步的次佳选择在 5 号位置，因为后手玩家除了下在四个角落外必败，即使

对手下了该位置，只要再下在对手的另一个对角也还有三分之二的胜算。

后手玩家的第1步，如果5号位置是空的则选择5号位置，否则选择在四个角落。

将 Tic-Tac-Toe 游戏棋盘扩宽到20路，将三连子获胜扩展到五连子获胜，就成了五子棋。五子棋的变化比 Tic-Tac-Toe 游戏要复杂得多，已经研究得知对黑方先手而言，有二十多种必胜开局。即只要黑方按照必胜开局行棋，必得获胜。为了规避这种先手的巨大优势，在很多比赛中就规定了一些先手的禁手，有“三、三”、“四、四”和“长连”等禁手。

在行棋过程中将五子棋的先后手不停的交换，即黑、白、白、黑、黑、白，一直下去，再将五连子获胜改成六连子获胜，就成了六子棋。六子棋中没有了先手的优势，到目前为止还没有发现必胜局面。

1.1.2 国际象棋机器博弈

讲到机器博弈，必然要提到国际象棋，因为它最早与计算机技术相结合，而且研究得最为深入的一种博弈游戏。

国际象棋棋盘是个正方形，由横纵各8格、颜色一深一浅交错排列的64个小方格组成。深色格称黑格，浅色格称白格，棋子就放在这些格子中移动。棋子共32个，分为黑白两组，各16个，由对弈双方各执一组，兵种是一样的，分为六种：王(1)、后(1)、车(2)、象(2)、马(2)、兵(8)。

在正式比赛中，国际象棋棋子采用立体棋子，非正式比赛中可以采用平面图案的棋子。

布子规则：上方16个为黑子，下方为白子。

```

车马象后王象马车
兵兵兵兵兵兵兵兵
□□□□□□□□
□□□□□□□□
□□□□□□□□
□□□□□□□□
兵兵兵兵兵兵兵兵
车马象后王象马车

```

“深蓝”与俄罗斯国际象棋大师卡斯帕罗夫的人机大战，使很多的中国人了解了国际象棋。“深蓝”是IBM专门为下国际象棋而设计的超级计算机，由256个信息处理器组成，每秒钟能做2亿次运算，平均每3分钟走一步棋。“深蓝”的程序代码是用C语言编写的并在AIX操作系统下运行。这种网络是可扩充的并行系统，它能在标准国际象棋比赛中为选手每步所分配的3分钟内计算出500亿至1000亿步。为了向“深蓝”提供可供利用的更为丰富的资源，“深蓝”小组收集了一种开局数据库，为该系统提供了象棋大师在过去100年间所进行的开局范例。除开局数据库外还应有终局数据库，终局数据库只在棋盘上仅剩五粒子时才被激活。该数据库提供数十亿场终局。同时，它还记录着本世纪以来国际象棋大师曾使用过的60万个棋谱。

1996年2月10日，美国费城，旷世棋王卡斯帕罗夫与IBM公司研制的“深蓝”(DEEP

BLUE) 计算机进行了震惊世界的国际象棋比赛。比赛期间, 全球每天有 2000 万人通过互联网观看赛事, 查询棋局。在这场举世瞩目的人与计算机的大战中, 卡斯帕罗夫显示了人类的无穷智慧, 以 4 比 2 的比分战胜了“深蓝”, 这个结果让世界再次对他叹服。

一年后, 美国纽约, 卡斯帕罗夫与计算机“深蓝”的升级版“更深的蓝”之间展开第 2 轮国际象棋人机大战。“更深的蓝”吸取了第 1 次人机大战的教训, 在开局、棋局的布置上更上一层楼。这一次, 卡斯帕罗夫输了, 他以 2.5 比 3.5 输给了“更深的蓝”。比赛结束后, 人们对他的表现感到大惑不解, 因为卡斯帕罗夫在至关重要的第 6 局决胜局中崩溃了, 他完全没有发挥出自己的正常水平, 仅下了 19 步就被杀得溃不成军, 输掉了比赛。这在过去是绝无仅有的。似乎是为了解释疑惑, 卡斯帕罗夫后来解释说: “我过于听从计算机专家的建议了, 没有发挥出自己的潜力和运用好战术, 这次我真的对计算机感到了畏惧。”

这场前无古人的人机大战给不仅是棋界, 包括整个人类都造成了深远影响。正是“深蓝”与卡斯帕罗夫的两番对决, 让更多的人认识了人工智能, 认识了机器博弈。“深蓝”虽然只是陪人类下棋的玩具, 它可以帮助人们研究如何最大限度地发挥计算机的优势, 将“深蓝”计算机中所涉及到的技术转而应用于其他领域, 更好地服务于人类。同样, 其他用于博弈的计算机的成果也将进一步应用甚至深入到我们目前所无法做到的领域; 人工智能的开发最终将会使人类受益。

1.1.3 机器博弈发展趋势

人工智能技术的不断发展, 博弈理论的推广, 计算机软硬件日新月异, 以及棋类游戏的广泛群众基础, 不断推动机器博弈的向前发展。

博弈棋种越来越多

在国际上开展最早也最广泛的博弈软件是国际象棋博弈软件, 在“深蓝”之前有“深思”, “弗里茨”等, 在“深蓝”之后有“更弗里茨”等博弈计算机。除国际象棋外, 越来越多的棋类游戏被纳入了机器博弈的范围。如: 中国象棋、围棋、9 路围棋、六子棋、日本将棋、亚马逊、西洋双陆棋、国际跳棋、幻影围棋、苏腊卡塔尔、四人陆战棋。

计算速度越来越快

计算机下棋严重依赖 CPU 运算速度。CPU 运算越快, 则可以搜索更多的可能局面, 搜索的深度可以更深, 下棋水平就越高。以目前世界上任何一台 PC, 要想在国际象棋上战胜卡斯帕罗夫, 简直就是梦想。“深蓝”完全依靠几百个 CPU, 超强的计算能力才与卡斯帕罗夫相抗衡。

CPU 处理速度正按照摩尔定理不断发展, PC 双核 CPU、四核 CPU 不再是新鲜事, 小型服务器上几十个处理器也越来越广泛, 拥有上千个处理器的超级计算机也越来越多的应用于各个领域。像“深蓝”那样有几百个 CPU 的超级专用博弈机器只有 IBM 这样的巨头才有财力研发, 普通软件开发商开发的博弈程序更多的是在小型服务器上运行, 而一般的程序设计员还只能在 PC 上进行开发。

1.2 中国象棋程序

中国象棋程序，也即中国象棋机器博弈，是一种能够自动下中国象棋的程序，根据当前棋局的状况，走一步合理的棋。象棋程序既可以与人下棋，也可以与另一个象棋程序进行较量。中国象棋程序在中国发展得比较晚，研究的人不多，了解的人也不多。

1.2.1 中国象棋博弈程序组成

中国象棋博弈程序主要由局面表示、走法表示及生成、局面评估、搜索算法、界面控制等5大部分组成。

局面就是一盘棋经过若干回合之后当前所处的形势，包括棋盘、红黑双方所剩棋子及其在棋盘上的分布、当前该走棋一方、双方所剩时间、双方所剩走棋步数等内容。局面表示是象棋程序的基础，局面表示的好坏直接关系到走法生成、局面评估和搜索算法的效率，从而影响象棋程序得到的最佳走法。

象棋程序每一次思考的目的是获取一个最佳走法（至少在程序看来是最佳的），要实现这一目的的简单方法就是生成全部所有可能的走法，然后再一个一个的比较，找出最佳的一个。实际上程序也是这样做的。

局面评估就是判断局面对红方（或者黑方，或者是当前走棋一方）的优势，并把优势进行量化。由于象棋程序搜索复杂度太大，搜索函数不可能搜索到棋局终了的状态，所以必须在某个深度的结点上结束并返回上一层，这个结点并没有达到棋局结束（胜平负），应该给它一个值，反映局面状况，对红方有利还是对黑方有利，有多少优势。必须把这种优势量化，以便于不同结点的优势可以比较，以确定哪一个结点更好。

中国象棋的局面变化实在是太多了，有时候一个局面可能走法达100多种，一般局面也有40多种走法。要完全搜索10步棋需要3.3年，即使完全搜索7步棋也要27分钟（按每秒搜索 10^8 个结点计）。按一盘棋平均100步（50个回合）计，要完全搜索100步是绝对不可能的。如何让计算机在有限的时间内搜索到更多的空间，搜索到更深的步数，是计算机博弈程序必须考虑的问题。这除了与计算机硬件有关之外，与搜索算法关系很大，这是因为在搜索树空间中有些分支是多余的，搜索的时候可以跳过某些分支。跳过的分支越多，搜索的速度越快，但漏掉最优解的可能性也在增加。搜索算法必须又快又准的找到最优解。

一般棋手不可能理解计算机内的数据结构，同样计算机也很难看懂普通的象棋棋盘，所以要实现人与计算机博弈程序的较量，必须要有一个中介，在人与计算机博弈程序之间传递信息。这个中介就是一个图形界面程序，它将局面及计算机走棋以图形界面展示给棋手，将棋手的走棋以特定数据结构传递给计算机博弈程序。博弈程序可以自带一个图形界面，它接收用户（棋手）的输入，即棋手所走的棋，然后将自己思考后所走的棋以图形显示在屏幕上，展示给棋手。

1.2.2 中国象棋程序研究现状

中国象棋程序的研究是从国际象棋程序中延伸而来，中国象棋与国际象棋有很大的相似