

Universidade Federal da Bahia
Universidade Estadual de Feira de Santana
Programa de Pós-Graduação em Ensino, Filosofia e
História das Ciências

Dissertação de Mestrado:

A Máquina e Seus Limites:
Uma Investigação Sobre o Xadrez Computacional

Cláudio Alves de Amorim

Orientador: Prof. Dr. Robinson Moreira Tenório

Salvador
2002

A Máquina e seus Limites:
Uma Investigação Sobre o Xadrez Computacional

Universidade Federal da Bahia
Universidade Estadual de Feira de Santana
Programa de Pós-Graduação em Ensino, Filosofia e
História das Ciências

Dissertação de Mestrado:

A Máquina e Seus Limites:
Uma Investigação Sobre o Xadrez Computacional

Cláudio Alves de Amorim

Orientador: Prof. Dr. Robinson Moreira Tenório

Dissertação apresentada ao Programa de Pós-Graduação em Ensino, Filosofia e História das Ciências da Universidade Federal da Bahia e Universidade Estadual de Feira de Santana, como requisito para a obtenção do título de Mestre

Salvador
2002

*À Mulher, Nelma, e
Às Filhas, Vanessa e Ana Cláudia,
companheiras de percurso,
com amor.*

*Ao Pai, Paulo,
À Mãe, Maria,
Ao Irmão, Paulinho, e
À Irmã, Rejani,
meus pontos cardeais,
com inexprimível gratidão.*

Agradecimentos

*Ao Prof. Robinson Tenório,
pela dedicação incomum
Ao Prof. João Carlos Salles,
pela fraterna erudição
Aos Profs. Olival Freire Jr.,
Oswaldo Pessoa Jr. e Charbel El-Hani,
pelo apoio permanente
Ao Prof. Harry Collins,
pelo diálogo generoso
Ao Amigo Marco Tek,
pelo amor às nossas crianças
Aos colegas e professores da saudosa
Escola Técnica Federal da Bahia e do
Clube Baiano de Xadrez,
pelos maravilhosos anos de convivência
(sem que eu soubesse, foi lá onde comecei
esta dissertação).*

Muito Obrigado

*“Ora, eu diria que um computador não é mais do que um lápis
endeusado. Portanto, derrubemos do pedestal não só o homem
como também, para variar, o computador”.*

Karl Popper

Sumário

| | |
|---|--------------------|
| <u>Lista de Figuras.....</u> | <u>vi</u> |
| <u>Lista de Diagramas.....</u> | <u>vii</u> |
| <u>Lista de Tabelas.....</u> | <u>viii</u> |
| <u>Resumo.....</u> | <u>ix</u> |
| <u>Abstract.....</u> | <u>x</u> |
| <u>1. Introdução.....</u> | <u>1</u> |
| 1.1. Os desafios da inteligência artificial..... | 1 |
| 1.2. O xadrez computacional..... | 5 |
| 1.3. Deep Blue x Kasparov..... | 9 |
| 1.4. Estrutura da dissertação..... | 11 |
| <u>2. Computadores e Computação.....</u> | <u>14</u> |
| 2.1. Algoritmos e programas..... | 14 |
| 2.2. Aspectos práticos da computação..... | 19 |
| 2.3. As máquinas e seu desempenho..... | 24 |
| <u>3. Jogos e Computadores.....</u> | <u>28</u> |
| 3.1. Terminologia..... | 28 |
| 3.2. Características dos jogos..... | 29 |
| 3.3. Os jogos como problemas de busca..... | 34 |
| 3.4. Os problemas da busca exaustiva..... | 38 |
| 3.5. Buscas otimizadas..... | 43 |
| 3.6. O xadrez como campo de provas..... | 47 |
| <u>4. Xadrez Computacional.....</u> | <u>52</u> |
| 4.1. Um pouco de história..... | 52 |
| 4.2. O estado da arte..... | 55 |
| 4.3. Problemas de desempenho..... | 65 |
| 4.4. A questão das heurísticas..... | 72 |
| 4.5. Eficácia x competência..... | 80 |
| <u>5. O Saber Algoritmizado.....</u> | <u>86</u> |
| 5.1. Um mundo de algoritmos..... | 86 |
| 5.2. Implicações pedagógicas..... | 90 |
| 5.3. Ferramenta epistemológica..... | 100 |
| <u>6. Conclusão.....</u> | <u>106</u> |
| <u>Apêndice I: Confrontos Homem-Máquina.....</u> | <u>112</u> |
| <u>Apêndice II: Partidas do Deep Thought e do Deep Blue.....</u> | <u>119</u> |
| <u>Apêndice III: Partidas do Hiarc8.....</u> | <u>125</u> |
| <u>Apêndice IV: Sistema Algébrico de Notação.....</u> | <u>126</u> |
| <u>Referências Bibliográficas.....</u> | <u>128</u> |

Lista de Figuras

| | |
|--|-----------|
| Figura 1: Exemplo de posição do jogo-da-velha..... | 37 |
| Figura 2: Árvore de busca com fator de derivação igual a 3..... | 39 |
| Figura 3: Escolha de um lance pelo algoritmo mini-max..... | 40 |

Lista de Diagramas

| | |
|---|-----------|
| Diagrama 1: Cravada permanente..... | 69 |
| Diagrama 2: Valor relativo de uma debilidade..... | 74 |
| Diagrama 3: Fácil para os humanos, difícil para os computadores..... | 83 |
| Diagrama 4: Zurique 1953 - Szabo x Petrosian..... | 93 |
| Diagrama 5: Zurique 1953 - Reshevsky x Averbach..... | 93 |

Lista de Tabelas

| | |
|---|-----------|
| Tabela 1: Instâncias do problema do caixeiro-viajante..... | 23 |
| Tabela 2: Alguns jogos clássicos..... | 32 |

Resumo

A inteligência artificial (IA) tenta realizar, por meio dos computadores eletrônicos digitais, tarefas que, quando realizadas por seres humanos, requerem raciocínio, conhecimento, julgamento e previsão. Ao longo de sua história, a IA tem feito promessas que não se cumpriram, mas que seguem se repetindo. Seus sucessos são normalmente alardeados, enquanto seus fracassos permanecem na penumbra, timidamente discutidos nos meios especializados. Deve-se ampliar e difundir as críticas à inteligência artificial, como forma de estimular o debate sobre o uso dos computadores, e sobre a destinação dos recursos que a IA recebe de instituições públicas e privadas. Na medida em que os computadores, cada vez mais, substituem pessoas em tarefas importantes, é preciso que seus limites e possibilidades sejam compreendidos.

Os jogos são um campo fértil para as críticas à inteligência artificial porque, ao contrário de problemas do “mundo real”, eles proporcionam as melhores chances para as máquinas, na comparação com a performance humana. Dentre os jogos, o xadrez tem sido um dos mais estudados, e o xadrez computacional tem uma tradição tão antiga quanto a própria computação eletrônica. Em 1997, o match-revanche entre o campeão do mundo, Garry Kasparov, e o computador Deep Blue, vencido pela máquina, causou sensação. Hoje, os programas baseados na combinação de heurísticas e poder computacional bruto, embora eficazes nas competições, nem de longe simulam a competência dos bons jogadores. Mesmo no plano competitivo, deve-se perguntar por que os atuais computadores, calculando milhões de variantes por segundo, ainda são superados pelos enxadristas de elite.

As dificuldades em se desenvolver máquinas que simulem a competência humana, em um domínio formal, como o xadrez, indicam que a imitação da razão, por meios computacionais, continua tão distante como sempre esteve. Por outro lado, as conquistas do xadrez computacional caracterizam os computadores como ferramentas epistemológicas, que nos auxiliam a ampliar os horizontes do conhecimento. Para que deles se faça o melhor uso, é preciso que a educação enfatize o juízo crítico, o diálogo e a inventividade.

Analisando os fundamentos do xadrez computacional, esta dissertação visa subsidiar a crítica a um dos dogmas prevalentes da IA: a idéia de que o software certo, rodando em um hardware suficientemente poderoso, pode imitar quaisquer aspectos da inteligência. Como alternativa à superação do homem pela máquina, propõe-se a idéia de computador como ferramenta epistemológica, destinada a servir à razão humana.

Palavras-chave: Inteligência Artificial; Xadrez; Computação; Epistemologia; Educação.

Abstract

Using electronic digital computers, Artificial Intelligence (AI) seeks to perform tasks that require knowledge, reasoning, judging and foresight when performed by humans. For about fifty years, AI has been repeating unfulfilled promises. Its achievements are readily boasted of, while its failures are kept in the shadows, as they are quietly discussed inside specialized circles. It is necessary to intensify the critiques of AI and make them broadly known, in order to promote debate over the use of computers, as well as over the funding provided to AI research, by both the public and private sectors. As computers increasingly replace people in important tasks, their limits and possibilities must be thoroughly understood.

In comparison to “real world” problems, games offer the best possibilities for machines to challenge human performance. Thus, they are most suitable as a workbench for the critiques of AI. Chess, among all games, is one of the most widely studied, and computer chess is as old as electronic computing itself. In 1997, the rematch between world champion Garry Kasparov and computer Deep Blue, won by the machine, generated huge interest. Nowadays, computer programs relying on a blend of heuristics and sheer calculating power are fearsome competitors, but they are nowhere near to simulating the competence of strong players. Even from a competitive standpoint, it may be asked why computers, which calculate millions of lines of play each second, are still beaten by elite chess players.

The fact that machines cannot simulate human competence, in a formal domain like chess, indicates that the duplication of human reason through computers is a far off goal as it has always been. On the other hand, the achievements of computer chess show that computers are great epistemological tools, which help us to broaden the horizons of knowledge. For computers to be used to their full potential, education must stress judging, dialog and inventiveness.

Through the analysis of computer chess, this work aims to strengthen the critique of a prevalent dogma of AI, namely the idea that the right software running on sufficiently powerful hardware can simulate any aspects of intelligence. The vision of the computer as an epistemological tool serving human reason is conveyed, as opposed to the overshadowing of man by the machine.

Keywords: Artificial Intelligence; Chess; Computing; Epistemology; Education.

1. Introdução

1.1. Os desafios da inteligência artificial

Gradativamente, o computador tem se infiltrado em todos os espaços sociais e, assim como o automóvel, condiciona o nosso modo de vida, de maneira tão sutil que passa despercebido. Entretanto, poucas pessoas entendem os computadores, e um número ainda menor percebe o quanto eles têm influenciado a história recente da humanidade.

Quando foram criados, no início dos anos 50, os computadores faziam cálculos balísticos e trabalhavam com os códigos secretos dos militares. Pouco tempo depois, passaram a processar os grandes volumes de dados envolvidos nas operações logísticas e transações financeiras dos grandes grupos privados e estatais. No início dos anos 80, com a produção em massa dos microprocessadores, os computadores subitamente se transformaram em instrumentos de uso pessoal e cotidiano, no trabalho e no lazer, sobretudo nos países ricos. Os sistemas de informação automatizados, cuja disseminação esbarrava nos altos custos e nas limitações operacionais dos equipamentos, passaram a ser amplamente utilizados, com profundos reflexos na organização socioeconômica dos povos. A chamada globalização, com todos os seus complicadores, está fortemente atrelada aos sistemas de informação em tempo real, que viabilizam o tráfego de informações e a movimentação de capital, a cada minuto, todos os dias do ano.

Muito antes da World Wide Web, Weizenbaum (1976, p.28) alertava para o fato de que “...a introdução do computador em algumas atividades humanas complexas pode se constituir em um compromisso irreversível.”¹ Evidentemente, esse vaticínio já se confirmou, e há muito as forças dominantes do capitalismo promovem uma relação instrumental com os computadores (saber usar, saber programar), visando tão somente a inserção dos indivíduos no mundo do trabalho computadorizado. Atualmente já não é mais possível conceber as atividades econômicas mais elementares, sem apoio dos computadores, seja ostensivamente, seja por vias indiretas. Mesmo nas regiões mais pobres, onde o computador não está presente, as pessoas têm suas vidas amplamente influenciadas pelos acontecimentos nos *mercados* internacionais, totalmente informatizados.

¹ “...the introduction of the computer into some complex human activities, may constitute an irreversible commitment.”

Em grande medida, foi graças às novas técnicas computacionais, desenvolvidas nos últimos vinte anos, que os Estados Unidos se consolidaram como potência militar hegemônica. Diferentemente do que ocorreu no Vietnã, as tropas americanas no Golfo Pérsico, em 1991, e no Afeganistão, em 2001, sabiam sempre a sua localização exata, graças ao uso do GPS – Global Positioning System. E, graças aos microprocessadores, os soldados agora enxergam perfeitamente à noite, usando binóculos infravermelhos de alta definição. Os mísseis, aviões e helicópteros totalmente computadorizados bombardeiam com *precisão cirúrgica* – um eufemismo para o massacre de inimigos indefesos, porque não dominam a tecnologia da informação. Hoje em dia vence a guerra quem projeta e fabrica circuitos integrados, programa o armamento pesado e gerencia a rede de satélites de comunicação, espionagem e localização de tropas. Nesse contexto, não é outra a lógica do programa “Guerra nas Estrelas”, do presidente George W. Bush: trata-se de equacionar, *por via computacional*, uma fragilidade do sistema de defesa norte-americano, representada pela suposta ameaça de mísseis nucleares inimigos.

Até os anos oitenta, a inteligência artificial permanecia confinada nos laboratórios de pesquisa, com raríssimas aplicações práticas. A partir dos anos noventa, com a evolução vertiginosa do hardware, a situação mudou radicalmente. Hoje em dia, a inteligência artificial está no cotidiano das pessoas, sob a forma de produtos de consumo, e também se faz presente em diversos sistemas de informação de uso comercial. A substituição dos seres humanos por dispositivos automáticos já ocorreu em certas tarefas críticas, que exigem a tomada de decisões – o que torna ainda mais significativas as questões que envolvem o uso (e potencial abuso) dos computadores.

Embora não exista uma definição única para o termo inteligência artificial (Russell e Norvig, 1995, pp.3-8), ele será adotado aqui para designar um campo de trabalho multidisciplinar, que tem como objetivo a construção de máquinas capazes de realizar tarefas que, quando realizadas por pessoas, requerem raciocínio, conhecimento, julgamento e previsão. Em função de seus objetivos especiais, a inteligência artificial (IA) é comumente apresentada como uma área autônoma, quando na verdade é uma aplicação de técnicas da ciência da computação.

Os sucessos da IA, em certos domínios restritos, são suficientes para que pesquisadores consagrados, a exemplo de Raymond Kurzweil (2000), apresentem o computador como uma máquina *sem limites*, capaz de protagonizar a solução de inquietantes problemas que afligem a humanidade. Extrapolando sobre os resultados obtidos até o presente, Kurzweil afirma que em breve os computadores serão capazes de superar todas as habilidades intelectuais dos humanos. Tais extrapolações, sem respaldo epistemológico,

somam-se a outros argumentos deficientes, como as comparações realizadas por Hans Moravec (1988, pp.57-61), entre o sistema nervoso humano e os circuitos eletrônicos digitais.

Kurzweil e Moravec, a exemplo de outros cientistas eminentes, defendem teses radicais, sem ao menos se dar conta de seus pressupostos filosóficos. Ao fazê-lo, lidam de forma simplista com conceitos complexos, tais como inteligência, crença, raciocínio, conhecimento e intenção, alterando arbitrariamente o sentido usual das palavras (Button e al., 1997; Searle, 1997, pp.36-38). A pretexto de cientificidade, esses pesquisadores endossam o *dogma do software certo no hardware suficientemente poderoso*, que segue dominante, malgrado as inúmeras promessas não cumpridas e previsões frustradas da inteligência artificial, nos seus cinquenta anos de história. Em linhas gerais, esse dogma – já criticado por Weizenbaum (1976) – traduz a idéia de que os algoritmos certos, rodando em máquinas com um poder computacional adequado, podem superar a inteligência humana, em todas as suas manifestações. Assim, diz Weizenbaum, sempre que falha uma previsão, os adeptos da IA a projetam para adiante, argumentando que, no futuro, a previsão se realizará, mercê dos avanços do hardware e do software². Ao contrário do que ocorre nas ciências, em geral, as previsões frustradas da inteligência artificial não costumam desencadear críticas aos programas de pesquisa vigentes, que continuam sendo custeados, ano após ano³.

Aceitar que determinados computadores são inteligentes poderia ser uma questão de escolha, de acordo com o tipo de significado que déssemos à palavra “inteligente”. Poderíamos dizer que o Deep Blue é inteligente, porque venceu o campeão mundial de xadrez, ou que uma calculadora de bolso é inteligente, porque faz cálculos precisos rapidamente, ou que um aparelho de ar-condicionado é inteligente, porque mantém a sala em uma temperatura estipulada. Em um caso extremo, o inventor do termo Inteligência Artificial, John McCarthy, afirma que “máquinas tão simples como termostatos têm – pode-se dizer – *crenças*” (apud Searle 1997, p.37 – grifo meu). Contudo, essa manipulação forçada da palavra inteligência e de suas correlatas simplesmente não esclarece nada.

Historicamente, metáforas derivadas dos paradigmas científicos emergentes, bem como das novidades tecnológicas de cada época, têm influenciado fortemente a forma como o ser humano vê a si próprio. A partir da década de 1940, com o advento da cibernética e

² As relações entre hardware e software são exploradas na seção 2.3.

³ Sou grato ao Prof. Harry Collins, da Universidade de Cardiff, por me chamar a atenção para este ponto extremamente importante.

da computação, a visão do homem como *máquina processadora de informações* ganhou força. Um dos pioneiros nessa linha, Warren McCulloch, é quem nos fala com eloquência, em um texto de 1955 (apud Dupuy, 1996, p.53-54):

Quanto mais aprendemos sobre os organismos, mais somos levados a concluir que eles não são simplesmente análogos às máquinas, mas *são máquinas* [...] As máquinas feitas pelas mãos do homem não são cérebros, mas os cérebros são uma variedade, muito mal-compreendida, de máquinas computacionais. A cibernética contribuiu para derrubar a muralha que separava o mundo magnífico da física do *gueto da mente* (grifos meus).

Segundo Dupuy, nem todos os ciberneticistas (hora rebatizados cognitivistas) defendem essa identificação ontológica entre cérebros e computadores, mas todos aceitam que existe uma identificação lógica entre ambos. Em outras palavras, pensar pode não ser necessariamente calcular, mas é sempre simular. Portanto, se os computadores podem simular, também podem, em princípio, pensar. Essa é a posição que sustenta os mais ambiciosos programas de pesquisa da inteligência artificial, na linha que, após Searle (1980), convencionou-se chamar de *IA Forte*.

Porém, uma coisa é saber que uma máquina realiza tarefas fora do alcance dos seres humanos, processando bilhões de operações lógicas e aritméticas por segundo, de acordo com um programa pré-estabelecido. Outra coisa é atribuir à máquina a capacidade de substituir o senso crítico, a responsabilidade e a imaginação das pessoas, por meios puramente mecânicos. Assim, entre a realidade dos computadores atuais e as promessas da inteligência artificial, há um abismo conceitual que nunca foi satisfatoriamente preenchido, e que não será superado por simples jogos de palavras. Ignorar esse fato implica em se alimentar expectativas fantasiosas sobre os problemas que os computadores podem resolver, em detrimento de uma discussão aprofundada do seu papel social.

Isaac Asimov produziu dezenas de histórias inspiradas na inteligência artificial (por exemplo, Asimov, 1991). Nelas, a convivência entre as pessoas e as máquinas inteligentes é quase sempre problemática, e freqüentemente, conflituosa – embora raramente destrutiva. Nos atuais livros, documentários e artigos de divulgação científica, contudo, as máquinas inteligentes são apresentadas basicamente como soluções para os problemas humanos, nunca como criadoras de novos problemas, que a humanidade terá que resolver. Essa inversão entre ficção e realidade reflete uma postura ingênua de cientistas e jornalistas, a benefício das grandes corporações de informática e

mídia, que lucram com o *fetichismo tecnológico* dos consumidores. Enquanto isso, escasseiam debates públicos, buscando esclarecer os pressupostos, objetivos, limites e possibilidades da inteligência artificial.

Quais são, afinal, os limites do computador? Já se tem, de fato, elementos que comprovem a possibilidade de simular a inteligência humana, por meio de algoritmos? Há boas razões para se crer no dogma dos algoritmos certos, rodando em hardware suficientemente poderoso? O que significa dizer que a máquina superou o ser humano em determinada atividade? São questões que envolvem problemas filosóficos que não devem ser ignorados. Conforme alertam Button e al. (1997), ainda que os sucessos da IA fossem proporcionais às promessas, o que de fato não ocorre, os problemas empíricos de ciência e engenharia não se sobreporiam aos problemas filosóficos, mas continuariam a conviver com eles. No campo especificamente epistemológico, por outro lado, a inteligência artificial precisa ter quebrada a sua *quase-imunidade*: para que o conhecimento avance, em qualquer área, as críticas e os fracassos precisam ser levados a sério.

Em um plano mais imediato, importantes decisões no mundo político, militar e empresarial passam pelas promessas da IA, que em função disso têm um impacto crescente na aplicação dos recursos disponíveis para a pesquisa e implantação de grandes projetos. Compreender os limites da máquina, portanto, não é apenas um objetivo teórico relevante: é também uma questão de cidadania, e um problema para a educação.

1.2. O xadrez computacional

Em última análise, os limites da inteligência artificial são os limites do que é realizável por meios computacionais. Sob essa perspectiva, os jogos constituem um interessante campo de análise. Ao mesmo tempo em que põem em destaque certos pontos fortes dos computadores, os jogos também nos permitem apreciar seus limites fundamentais. Além disso, o apelo lúdico dos jogos os transforma em um tópico ainda mais atraente. O xadrez, que é um modelo de jogo computacional, tem uma série de vantagens, enquanto campo de investigação dos limites das máquinas: (1) Uma tradição de disputas entre máquinas, e entre seres humanos e máquinas, com inúmeras partidas anotadas; (2) Uma vasta bibliografia, que proporciona referenciais para a análise da capacidade das máquinas jogadoras; (3) Um forte apelo cultural, reunindo adeptos numerosos e entusiasmados, em todo o mundo.

Em alguns jogos, como o Go⁴ e o pôquer, o desempenho dos melhores programas ainda está muito aquém do desempenho dos melhores jogadores humanos. Em outros, como as damas, o poder de cálculo e capacidade de armazenamento dos computadores permite a implementação de programa quase perfeitos, de forma que os humanos já não têm chance contra as máquinas. Também o gamão é jogado pelos computadores quase com perfeição, mas por ser um jogo relativamente simples, os melhores jogadores humanos também têm um desempenho quase perfeito (Ginsberg, 1998). No xadrez, temos uma situação peculiar: *força bruta*⁵ e sofisticação heurística se combinam para a obtenção de um desempenho notável das máquinas, mas nada que se pareça com a perfeição relativa alcançada em jogos mais simples. Por isso, os melhores jogadores humanos continuam sendo adversários à altura para os computadores. É essa, justamente, uma das questões fundamentais que a presente dissertação visa esmiuçar: Por que, apesar de toda a tecnologia de hardware e software, bem como o esforço de enxadristas de sucesso, a serviço do xadrez computacional, os jogadores humanos de elite continuam desafiando as máquinas?

Antes de prosseguir, devemos reconhecer no xadrez computacional um campo de investigação pertinente, para a análise dos limites e possibilidades da inteligência artificial, já que essa não é uma posição unânime. John McCarthy (2002a), por exemplo, argumenta que as pesquisas no xadrez computacional, ao privilegiar a obtenção de resultados competitivos, através da força bruta, deixam de contribuir significativamente para o desenvolvimento da IA. Referindo-se ao alto padrão de jogo do Deep Blue, ele afirma que

[...] o fato de que este nível de jogo requeira milhões de vezes mais cálculos do que os realizados por um jogador humano é uma medida do nosso entendimento imperfeito dos princípios da inteligência artificial. Além disso,

⁴ O Go é um jogo oriental, extremamente difundido no Japão. É jogado sobre um tabuleiro onde linhas verticais e horizontais se cruzam, formando uma matriz quadrada de ordem 19. De modo semelhante às damas, o Go tem apenas um tipo de peça, divididas em duas cores, e ao contrário das damas e do xadrez, em que as peças são posicionadas nas casas do tabuleiro, as peças do Go são colocadas nas interseções das linhas. A tradição milenar, e a estratégia sofisticada fazem do Go um jogo desafiador, tanto para os praticantes como para os programadores.

⁵ Força bruta é um substantivo que qualifica os métodos computacionais que lançam mão do poder de processamento das máquinas para resolver problemas através de cálculos exaustivos.

o direcionamento dos trabalhos em xadrez computacional, em sua maioria, para o sucesso em jogos de torneio, teve um preço, em termos científicos.⁶

Ou seja, McCarthy faz uma nítida diferença entre a capacidade competitiva dos programas e a sua *inteligência*, medida pela capacidade de imitar o raciocínio humano, ou mais especificamente, pela possibilidade de jogar bem o xadrez, sem recorrer a cálculos exaustivos. Mais adiante, no mesmo artigo, o pioneiro da IA faz uma proposta radical: “os torneios deveriam admitir apenas programas com limites severos em termos de capacidade computacional. Isso faria com que a atenção [dos pesquisadores] se concentrasse nos avanços científicos”.⁷ Fica claro, assim, que a “ciência” a que ele se refere ocupa-se com a tentativa de imitar os caminhos do raciocínio humano, por via computacional. Para McCarthy, portanto, a vitória do Deep Blue sobre Kasparov foi a culminação de um “bem-sucedido esforço de engenharia, que durou 22 anos”, mas não uma vitória da Inteligência Artificial, em sentido estrito.

Conforme o próprio McCarthy (2002b) faz notar, contudo, a Inteligência Artificial trabalha com duas abordagens distintas: uma, que chamaremos de abordagem *cognitivista* enfatiza a simulação do raciocínio humano; a outra, que chamaremos de abordagem *pragmática*, consiste em usar os métodos computacionais disponíveis, explorando o poder de cálculo dos computadores para atingir determinados fins, sem a preocupação de imitar o raciocínio humano. Embora ambas as abordagens quase sempre andem juntas, na prática, a ênfase conferida a cada uma é que dá o tom das pesquisas. Os proponentes da via cognitivista buscam modelos para imitar a forma como o ser humano resolve problemas; os defensores da via pragmática enfatizam as soluções empíricas *ad hoc*, aplicando as teorias e técnicas que se mostrem úteis para os seus fins, sem a preocupação de acompanhar os caminhos do pensamento humano.

A despeito das objeções, há pelo menos três pontos que acentuam a importância do xadrez para a IA: (1) alguns autores, como Ford e Hayes (1998) consideram legítimo o desenvolvimento de técnicas computacionais para a superação das habilidades intelectuais humanas, com o recurso à força bruta, ou outros mecanismos alheios ao modo como os humanos pensam. Aliás, essa é a abordagem adotada por consagrados livros-texto da IA, como Rich e Knight (1991), Russel e Norvig (1995) e Arkin (1998).

⁶ “[...] it is a measure of our limited understanding of the principles of artificial intelligence that this level of play requires many millions of times as much computing as a human chess player does. Moreover, the fixation of most computer chess work on success in tournament play has come at a scientific cost”.

⁷ “tournaments should admit programs only with severe limits on computation. This would concentrate attention on scientific advances”.

Assim, a suposta “inteligência” das máquinas estaria nos fins atingidos, e não nos meios usados para atingir esses fins; (2) as máquinas de xadrez, por maior que seja o seu poder de cálculo, não vencem os humanos apenas com base na força bruta, mas necessitam de heurísticas sofisticadas para a avaliação posicional do jogo. Por essa via, os programas de xadrez estão, sim, irremediavelmente ligados à inteligência artificial, que têm no estudo das heurísticas um dos seus focos principais; (3) mesmo desconsiderando este último ponto, se concordarmos com a posição de McCarthy, devemos então perguntar por que, ao invés da força bruta, os pesquisadores do xadrez computacional não têm privilegiado outras técnicas, que aproximem seus programas do modo de pensar dos grandes mestres humanos.

As competições entre programas de xadrez, e entre programas e seres humanos, têm décadas de tradição. Durante todo esse tempo, portanto, o estímulo para o desenvolvimento de novas técnicas de IA aplicadas ao xadrez têm se multiplicado. Porém, apenas os programas desenvolvidos dentro da abordagem pragmática têm se demonstrado bem-sucedidos nos jogos de competição, de modo que essa abordagem tornou-se, de fato, dominante – o que até certo ponto justifica a crítica de McCarthy. De certo modo, contudo, a crítica de McCarthy tem um caráter anacrônico porque, até poucos anos atrás, os programas baseados em força bruta eram muito fracos, em comparação com bons jogadores humanos, e ainda assim, não surgiram técnicas alternativas convincentes. Então, seguindo uma via oposta à de McCarthy, poderíamos sugerir que o próprio fato de a abordagem pragmática ter se tornado hegemônica indica uma inerente dificuldade em se simular a competência enxadrística humana, por via cognitivista.

Por fim, devemos considerar que, malgrado a enorme força competitiva dos atuais programas de xadrez, há problemas que ainda não foram satisfatoriamente resolvidos. Especificamente, o tratamento que os programas dão a certas posições é muito primário, em comparação com o seu sucesso nas competições⁸. Há posições que os melhores programas são incapazes de avaliar satisfatoriamente, ou o fazem após calcular milhões de variantes, enquanto jogadores humanos de nível mediano as avaliam de imediato, sem nenhuma necessidade de cálculo. Portanto, a inteligência artificial continua sendo desafiada a encontrar métodos gerais para corrigir esse disparate, ou seja, métodos que não se apliquem apenas a posições específicas, e que sejam capazes de simular nas máquinas a *compreensão* do xadrez, nos moldes humanos.

⁸ A esse respeito, ver os comentários às partidas do Apêndice I, bem como o exemplo de avaliação posicional distorcida, discutido na seção 4.5.

1.3. Deep Blue x Kasparov

Em menos de trinta anos, os computadores de xadrez evoluíram de um nível comparável ao de humanos principiantes para um nível comparável ao dos maiores jogadores profissionais. Em partidas rápidas (“xadrez-relâmpago”), os grandes programas comerciais da atualidade derrotam facilmente todos os humanos, exceto uns poucos, de talento excepcional. Em partidas de longa duração, em que a vantagem dos computadores é menor, estes mesmos programas representam um sério desafio para os cinquenta melhores jogadores no ranking da FIDE⁹, o que fica bem caracterizado em algumas das recentes disputas envolvendo homens e máquinas. Com isso, programas instalados em microcomputadores amplamente disponíveis alcançam um *rating performance*¹⁰ só inferior ao de algumas dezenas de enxadristas profissionais.

A força dos jogadores eletrônicos foi ratificada no memorável match¹¹ de 1996, quando o então campeão mundial de xadrez, Garry Kasparov, derrotou o computador Deep Blue, da IBM, pelo score de 4 a 2. Apesar do placar final favorável a Kasparov, a primeira partida, vencida pelo Deep Blue, causou forte impressão, tanto nos meios especializados como no enorme público que acompanhou o desenrolar do match: pela primeira vez, um computador derrotava um campeão mundial de xadrez, em uma partida jogada em condições tradicionais de torneio¹². O impacto maior, porém, ainda estava por vir: na revanche de 1997, uma nova versão do Deep Blue derrotou Kasparov,

⁹ FIDE (Fédération Internationale des Échecs) é a Federação Internacional de Xadrez, cujo ranking reflete o desempenho dos jogadores nos torneios oficiais ao redor do mundo.

¹⁰ O rating é um valor numérico que define a posição do jogador no ranking da FIDE, ou de qualquer outra federação de xadrez. O rating aumenta ou diminui, de acordo com os resultados do jogador nas competições, levando-se em conta o rating médio dos adversários com que ele se defronta. O rating performance é um rating presumível e não oficial, calculado com base nos resultados de uma máquina ou de um jogador, em uma série (normalmente reduzida) de partidas. O rating performance deve ser visto com reservas, porque apenas um número grande de resultados, em jogos contra muitos adversários diferentes, pode dar uma medida confiável da força de um jogador.

¹¹ No xadrez, dá-se o nome de match a uma série ininterrupta de jogos entre dois jogadores, ou seja, uma série não intercala pela participação dos jogadores em outras competições – o match, em si mesmo, é uma competição.

¹² Isto é, uma partida individual de longa duração, em contraste com as partidas rápidas e simultâneas, em que os computadores já haviam derrotado o próprio Kasparov, bem como outros ex-campeões mundiais. Em geral, são consideradas partidas “clássicas” aquelas em que cada jogador dispõe de pelo menos três minutos, em média, para a realização de cada lance.

pelo placar mínimo de 3,5 a 2,5 (duas vitórias, três empates e uma derrota). Com isso, a extraordinária performance das máquinas jogadoras de xadrez ficou definitivamente demonstrada. A fim de se avaliar melhor esse acontecimento, vale lembrar que Kasparov não era apenas o campeão do mundo na época: ele é considerado, por muitos dos seus pares, o maior enxadrista de todos os tempos, e sua hegemonia no cenário enxadrístico internacional segue firme desde 1985, quando, aos 22 anos, arrebatou o título mundial do seu compatriota russo Anatoly Karpov.

Ao terminar com a vitória da máquina sobre um jogador da estatura de Kasparov, o match de 1997 configura-se como um marco científico e tecnológico, cujas implicações merecem ser analisada em profundidade. Como a capacidade de jogar bem o xadrez sempre foi vista como um símbolo de inteligência, o público que acompanhou os confrontos entre Kasparov e o Deep Blue se perguntou como o computador pôde vencer o campeão do mundo. O próprio Kasparov, algo surpreso, afirmou ter vislumbrado traços de “inteligência” na máquina, em função da sua forma de jogar.

Atingido seu objetivo – derrotar o campeão mundial de xadrez –, a IBM abandonou o Deep Blue, e a equipe responsável pelo projeto se dissolveu. Nem antes nem depois dos matches com Kasparov o Deep Blue se defrontou com outros enxadristas de elite, embora desafios nesse sentido tenham surgido, inclusive por parte do indiano Vishwanatan Anand, atualmente o terceiro no ranking da FIDE, e a quem Kasparov considera um dos maiores, dentre os grandes jogadores que nunca chegaram ao título mundial. Desinteressando-se por esses desafios, a IBM nunca permitiu que a capacidade do Deep Blue fosse aferida nas mesmas condições a que se submetem os enxadristas humanos, ou seja, jogando contra adversários variados, com curtos espaços de tempo para rever estratégias entre uma partida e outra. Restam-nos, então, os dois confrontos Kasparov x Deep Blue que, no cômputo geral, o humano venceu por 6,5 a 5,5. O que significam esses números? (1) O Deep Blue era inteligente? (2) O Deep Blue era um jogador de xadrez tão bom quanto Kasparov? (3) Ambas as coisas? ou (4) Nenhuma das anteriores?

O Deep Blue usado no segundo encontro com Kasparov analisava de 100 a 200 milhões de posições por segundo, em alguns casos chegando a mais de 300 milhões (Campbell e al., 2002). Em uma partida em que cada jogador tem duas horas para realizar quarenta lances, esse poder de cálculo proporcionava ao Deep Blue a capacidade de analisar dezenas de bilhões de posições, antes de realizar um lance, e outras tantas, durante os períodos em que Kasparov preparava cada jogada. Em contraste, estima-se que os jogadores humanos de primeira linha consideram no máximo cem, talvez duzentas

posições, para se decidirem sobre cada lance, mas, na maioria das vezes, bem menos do que isso. Diante desse contraste, deve-se substituir a pergunta mais popular, “como a máquina pode derrotar o ser humano?”, por sua inversa, ou seja, como é possível que um ser humano ainda derrote a máquina em tais circunstâncias? Como é possível que, em determinado momento de uma partida, o computador analise, digamos, 50 bilhões de posições, apenas para constatar que todas as continuações disponíveis são perdedoras?

Contrastes à parte, persiste o desempenho extraordinário do Deep Blue. O que ele nos diz? Ryle (1984, p.45), afirma que “Ao julgar se a performance de alguém é inteligente ou não, nós temos, de certa forma, que olhar além da performance propriamente dita”.¹³ Deixando de lado a questão da inteligência, em sentido mais amplo, é possível que, para saber se alguém (ou alguma coisa) é um grande jogador de xadrez, seja necessário enxergar além das partidas propriamente ditas. Poderíamos, assim, perguntar se o modo como as máquinas avaliam posições pode ser considerado como um passo para o desenvolvimento de um *inteligência enxadrística*. Especificamente, dada a força demonstrada pelo Deep Blue no jogo competitivo, poderíamos considerá-lo um jogador de xadrez inteligente?

Ao lidar com estas questões, levaremos em consideração que, independente do inegável sucesso frente a Kasparov, o Deep Blue, assim como seu antecessor, o Deep Thought, tiveram momentos muito ruins, sofrendo derrotas frente a humanos e máquinas que dificilmente teriam chances contra o campeão do mundo (em especial a partida contra o microcomputador Mephisto, reproduzida no Apêndice II). *Olhando além da performance*, como quer Ryle, tentaremos entender o que as derrotas e os lances pobres do computador nos dizem a respeito dos seus limites, enquanto jogador de xadrez.

1.4. Estrutura da dissertação

O plano geral desta dissertação é investigar os limites e possibilidades dos computadores, usando o xadrez como campo de análise. O xadrez é um domínio estritamente formal, onde os computadores têm a possibilidade de demonstrar toda a sua força, e dissimular suas debilidades, ao simular a inteligência humana. Portanto, os limites eventuais das máquinas, enquanto jogadoras de xadrez, são indícios importantes para o entendimento desses limites em contextos mais abrangentes.

¹³ “In judging that someone’s performance is or is not intelligent, we have, in a certain manner to look beyond the performance itself”.

Comparados aos seres humanos, os computadores calculam com enorme velocidade, armazenam e recuperam um vasto volume de dados com exatidão, e além do mais, nunca se cansam e nem são afetados pelas emoções negativas. Mesmo assim, vários enxadristas – inclusive de segunda linha – continuam empatando e vencendo partidas contra os mais modernos programas, capazes de avaliar mais de um milhão de posições por segundo. Será que essa *resistência* dos enxadristas de talento, diante da força das máquinas, é apenas um problema circunstancial, ou será que ela nos revela algo a respeito dos limites da máquina, na simulação de capacidades humanas? A velocidade crescente do hardware e o aprimoramento dos algoritmos conduzirão a uma vantagem conclusiva dos computadores sobre os enxadristas humanos? Ainda que a superação definitiva dos humanos pelas máquinas ocorra, nas competições, qual será o seu significado? Terão os pesquisadores avançado rumo a máquinas capazes de simular habilidades humanas em contextos mais amplos? As mesmas técnicas que permitem aos computadores jogar xadrez com tanta eficácia se aplicam, por exemplo, ao aprendizado e ao ensino do jogo? O que as diferenças entre a abordagem humana e a abordagem computacional do xadrez podem nos dizer sobre o potencial do computador, enquanto *ferramenta epistemológica*¹⁴? Este é, em resumo, o ponto de partida desta investigação, que tem a seguinte estrutura:

No Capítulo 2, “Computadores e Computação” discute-se o computador eletrônico digital, tanto em termos lógicos como em termos tecnológicos. Analisa-se, também, os conceitos de algoritmo e programa, como base para a compreensão dos problemas apresentados no decorrer do texto.

No Capítulo 3, “Jogos e Computadores”, é formalizado o conceito de jogo utilizado neste trabalho. É discutido o tratamento computacional dos jogos, introduzindo-se o problema da complexidade computacional, e a tentativa de resolvê-lo através das heurísticas. Ao final do capítulo, discute-se o que faz do xadrez um jogo particularmente relevante para a análise dos limites e possibilidades do computador.

¹⁴ O termo ferramenta se aplica a artefatos que auxiliam as pessoas a realizar tarefas, estendendo as capacidades do organismo humano. A ferramenta é guiada pelo talento e pela razão, e portanto não substitui o homem. Sob esta perspectiva, o computador é uma ferramenta epistemológica, na medida em que assiste o ser humano na realização de tarefas mentais que ampliam decisivamente os horizontes do conhecimento. Em contraste com os instrumentos de pesquisa tradicionais, como telescópios, cronômetros e aceleradores de partículas, os computadores possibilitam a mecanização de certos procedimentos lógicos e aritméticos, viabilizando a investigação de *espaços abstratos* antes impenetráveis, porquanto estranhos aos procedimentos empíricos tradicionais.

O capítulo 4 chama-se “Xadrez Computacional”. Nele são analisadas as técnicas usadas pelos programas de xadrez que têm obtido os melhores resultados, ou seja, aqueles que são jogadores eficazes, sem a preocupação de imitar a forma como os humanos pensam. A questão das heurísticas é analisada em detalhes e, por fim, faz-se a diferenciação entre *performance competitiva* e *competência enxadrística*, mostrando-se que as máquinas, malgrado sua performance, não são jogadoras de xadrez competentes.

“O Saber Algoritmizado” é o título do capítulo 5, que começa com a análise da presença dos algoritmos na vida social, independentemente dos computadores. Depois, são discutidas certas conseqüências pedagógicas da crescente *algoritmização* do saber, em um contexto onde os computadores muitas vezes são vistos como modelos de eficácia. Finalmente, o computador é apresentado como uma ferramenta epistemológica, que contribui decisivamente para a ampliação dos horizontes do conhecimento, em diversas áreas do saber.

Na Conclusão, são recapitulados os pontos focais desta investigação, como contribuição para uma crítica às ambições e promessas da inteligência artificial.

O apêndice I apresenta exemplos específicos das limitações do xadrez computacional, através da análise detalhada de seis partidas, disputadas entre jogadores humanos e alguns dos mais fortes programas.

O apêndice II é uma coletânea de partidas do Deep Thought e do seu sucessor, o Deep Blue, jogadas entre 1988 e 1997. Além do interesse histórico, essa coletânea evidencia os desníveis no desempenho daquelas que foram as mais poderosas máquinas de xadrez jamais construídas.

O Apêndice III apresenta quatro partidas recentes do programa Hiarc 8, contra dois grandes mestres humanos, como exemplo do sucesso competitivo de um programa comercial, cujo autor prioriza a sofisticação heurística, em relação à velocidade no cálculo de variantes.

O Apêndice IV é uma explicação do sistema algébrico de notação para partidas de xadrez, usado neste trabalho.

2. Computadores e Computação

2.1. Algoritmos e programas

Em linhas gerais, algoritmo é um procedimento passo a passo, para se atingir determinado fim, descrito de forma que possa ser repetido várias vezes. Nesse sentido, uma receita culinária, assim como um roteiro teatral, são algoritmos. Contudo, no contexto desta dissertação, necessitamos de um conceito de algoritmo mais estrito, que sirva de base para a compreensão dos programas de computador. Os algoritmos que nos interessam são aqueles voltados para a solução de problemas descritos matematicamente.

Suponhamos que se queira somar dois números inteiros positivos, com até dez algarismos cada um, usando lápis e papel. Normalmente, escrevem-se um número sob o outro, e somam-se os pares de algarismos de mesma ordem, da direita para a esquerda, até completar toda a soma. Quando a soma de dois algarismos é maior do que dez, transporta-se o valor 1 para a soma dos algarismos de ordem imediatamente superior. Esse processo, que as pessoas realizam automaticamente, exprime-se nos seguintes passos:

1. Escrever no papel os dois números a serem somados, um sob o outro, alinhados pela direita.
2. Abaixo dos dois números, desenhar um traço reto. O resultado da operação aparecerá abaixo do traço.
3. Se um número possuir menos algarismos do que outro, as casas à esquerda do menor número serão preenchidas com zeros, até completar o maior número de algarismos.
4. Percorrer os números da direita para a esquerda, somando os pares de algarismos que se encontram um sob o outro. Enquanto houver algarismos a somar, faça:
 - 4.1. Encontrar na tabuada o resultado da soma
 - 4.2. Se o resultado for menor do que 10
 - 4.2.1. Então
Escrevê-lo abaixo dos algarismos somados
 - 4.2.2. Senão
Escrever o último algarismo do resultado obtido, abaixo dos algarismos somados
Transportar uma unidade para ser adicionada ao resultado da próxima soma
5. Se veio uma unidade da última soma realizada
 - 5.1. Então escreva o algarismo 1 à esquerda do resultado já obtido

Os passos de 1 a 3 se destinam apenas a arrumar os números; os passos 4 e 5, com seus respectivos sub-passos, se destinam à realização da soma, propriamente dita.

Inconscientemente, as pessoas dividem a operação original em operações mais elementares, que são executadas repetidamente, até que a soma maior esteja concluída: ao invés de somar os números de uma vez, a soma é feita algarismo a algarismo, de acordo com a tabuada que se aprende na escola.

Os passos acima descritos caracterizam um algoritmo, que é uma seqüência ordenada de instruções elementares, destinada a atingir um objetivo específico. O que chamamos de instrução elementar é uma operação que pode ser realizada pelo processador do algoritmo de uma só vez, isto é, em um só passo. No exemplo em foco, o *processador* é a pessoa que faz a soma, e a principal instrução elementar, que se repete até dez vezes, é a soma de dois algarismos, que a pessoa faz de cor, em função de ter decorado a tabuada de somas de 0 a 9.

No passo 4, a pessoa tem que verificar, após cada par de algarismos somados, se a soma deve parar ou prosseguir. No passo 4.2, ela deve verificar se há ou não há um transporte (“vai um”), e se houver, acrescentar a unidade à próxima soma. Finalmente, no passo 5, verifica-se se há um último transporte, após realizadas todas as somas individuais. Assim, os passos 4, 4.2 e 5 caracterizam o conceito de desvio condicional, que é uma alteração dos rumos do algoritmo, de acordo com condicionantes lógicos avaliados durante o processamento.

A partir desse exemplo, podemos sistematizar assim o conceito de algoritmo: algoritmo é uma seqüência ordenada de instruções elementares, passível de ser executada por um processador, e que atua sobre um conjunto de dados, com um objetivo bem definido. Os elementos fundamentais de um algoritmo são, portanto, o autor, o processador, os dados e o conjunto de instruções, juntamente com a linguagem em que se exprimem. No algoritmo de soma aqui descrito, os dados são os números a serem somados, o processador é a pessoa que faz a soma, a linguagem é a língua portuguesa, as instruções são os passos de 1 a 5, e o autor, é anônimo.

Vale ressaltar que o conceito de instrução elementar não é absoluto, e sim relativo à capacidade do processador. De fato, ao somar dois números com até dez algarismos, nada nos impede de somarmos os algarismos em grupos de dois ou três pares por vez, exceto o fato de termos decorado apenas a tabuada de somas de 0 a 9, ao invés de uma tabuada estendida, digamos, de 0 a 99, ou mesmo de 0 a 999. Se pudéssemos fazer a soma usando uma tabuada estendida, terminaríamos a operação mais rapidamente¹⁵. Por

¹⁵ De modo geral, quanto mais poderosas as operações elementares realizadas pelo processador, mais rapidamente ele será capaz de executar algoritmos. Alguns aspectos técnicos dessa importante questão

outro lado, para uma criança que ainda não sabe tabuada, a soma de cada par de algarismos é feita com base em um algoritmo, ao invés de se constituir em uma operação elementar: primeiro, a criança representa nos dedos um algarismo, depois, conta o número de dedos correspondente ao segundo algarismo, etc. Logo, o fato de uma operação ser elementar, ou não, depende da capacidade do processador que a executa.

No âmbito propriamente matemático, um dos algoritmos mais conhecidos, o algoritmo de Euclides, para achar o máximo divisor comum (MDC) entre dois números, foi originalmente concebido por volta do ano 300 a.C. (Penrose, 1991, p.32). Para explicar esse algoritmo, definamos a operação $\mathbf{a} \text{ MOD } \mathbf{b}$, como sendo o resto da divisão de \mathbf{a} por \mathbf{b} . Para encontrar o MDC entre \mathbf{a} e \mathbf{b} , com base no algoritmo de Euclides, procedemos da forma seguinte (supondo $\mathbf{a} \geq \mathbf{b}$; $\mathbf{a} > 0$ e $\mathbf{b} > 0$):

```
Faça  $\mathbf{c} \leftarrow \mathbf{a} \text{ MOD } \mathbf{b}$   
Enquanto  $\mathbf{b} > 0$   
    Faça  $\mathbf{a} \leftarrow \mathbf{b}$   
    Faça  $\mathbf{b} \leftarrow \mathbf{c}$   
    Faça  $\mathbf{c} \leftarrow \mathbf{a} \text{ MOD } \mathbf{b}$   
Imprima o valor de  $\mathbf{b}$ 
```

Nessa descrição do algoritmo, estamos admitindo, implicitamente, que a operação MOD é elementar. Mas, e se o processador usado para executar o algoritmo não fosse capaz de fazer divisões, mas apenas somas e subtrações? Nem por isso estaríamos impedidos de rodar o algoritmo de Euclides, bastando adaptá-lo à capacidade limitada do processador. Construiríamos, para tanto, um segundo algoritmo, equivalente à operação MOD:

```
Enquanto  $\mathbf{a} \geq \mathbf{b}$   
    Faça  $\mathbf{a} \leftarrow \mathbf{a} - \mathbf{b}$   
 $\mathbf{r} \leftarrow \mathbf{a}$   
( $\mathbf{r}$  é o resto da divisão de  $\mathbf{a}$  por  $\mathbf{b}$ )
```

Depois, seria só introduzir o segundo algoritmo no primeiro, nos lugares em que aparece a operação MOD.

A exemplo do que ocorre com a maioria dos problemas, não existe apenas um algoritmo para o MDC. Se não conhecêssemos o algoritmo de Euclides, poderíamos encontrar o MDC entre dois números usando uma abordagem do tipo “tentativa e erro” (ainda supondo $\mathbf{a} \geq \mathbf{b}$; $\mathbf{a} > 0$ e $\mathbf{b} > 0$):

são discutidos adiante (seção 2.3), no contexto dos computadores eletrônicos digitais.

```
Faça d ← b DIV 2
Enquanto ((a MOD d) <> 0) ou ((b MOD d) <> 0)
    Faça d ← d - 1
Imprima d
```

Aqui, o valor inicial de **d**, que é a metade de **b**, é subtraído de uma unidade por vez, até que **d** seja, ao mesmo tempo, divisor de **a** e de **b**. Na pior das hipóteses, o programa pára quando **d** for igual a um.

Este último algoritmo é logicamente equivalente ao de Euclides, pois chega ao mesmo resultado, mas é inferior em termos de desempenho, porque realiza um maior número de passos, até chegar à solução. Em termos puramente lógicos, a eficiência de um algoritmo não é relevante, desde que possamos demonstrar que ele proporciona uma solução satisfatória para o problema que se quer resolver. Contudo, a questão da eficiência é crucial na prática, porque determinada a viabilidade ou inviabilidade de se resolver certos problemas, por via computacional. Este ponto será detalhado na próxima seção, e retomado no capítulo 3, que discute a abordagem computacional dos jogos.

Antes de prosseguir, devemos estabelecer a diferença entre algoritmo e programa de computador, já que todo programa implementa um algoritmo, mas nem todo algoritmo se traduz em um programa. Ao criar algoritmos, podemos definir arbitrariamente o conjunto de instruções elementares que nos convém, bem como a precisão dos dados a serem usado, e ainda, a quantidade de dados que queremos processar. Ao fazer um programa, contudo, devemos levar em consideração as características do computador onde ele será executado¹⁶: qual é o espaço disponível para armazenamento de dados, quais são as instruções elementares que a máquina implementa, e qual é a precisão numérica com que ela é capaz de operar.

Além disso, o programa é escrito em uma linguagem de programação, estritamente formal. Sendo assim, o código do programa não admite ambigüidades, pois o computador não o interpretará, mas o executará os comandos inexoravelmente, quer sejam pertinentes ou não. Voltando ao exemplo da soma com lápis e papel, nosso primeiro passo nos diz para escrevermos os números, um abaixo do outro, alinhados à direita. Mas à direita em relação a quê? À folha de papel? À parede da sala? Claro que não; queremos os números alinhados à direita em relação um ao outro. Contudo, é o bom senso do *processador humano* que permite esse tipo de interpretação.

¹⁶ Em computação, costuma-se dizer que os computadores “rodam” programas, por analogia entre o movimento da roda e a movimentação dos sinais elétricos no interior dos circuitos.

Brassard e Bratley (1996) associam a propriedade de não-ambigüidade aos algoritmos, de modo geral, e não apenas aos programas:

A execução de um algoritmo não deve envolver decisões subjetivas, nem pode demandar o uso de intuição e criatividade. Assim, uma receita culinária pode ser considerada um algoritmo se ela descrever precisamente como preparar um prato, dando as quantidades exatas [de ingredientes] a utilizar, e instruções detalhadas sobre o tempo de cozimento. Por outro lado, se a receita incluir noções vagas, tais como “adicionar sal a gosto” ou “cozinhar até que fique macio”, então nós não a chamaríamos mais de algoritmo (p.2).¹⁷

Contudo, adicionar sal a gosto é um comando perfeitamente válido para uma pessoa, e não há problema algum em incluí-lo, se o algoritmo for executado por um “processador” humano. Voltando ao algoritmo de Euclides, se uma pessoa versada em matemática for encontrar o MDC entre dois números, e observar que **a** é menor do que **b**, ela inverterá os valores, antes de prosseguir. Um computador, ao contrário, alheio a essa nuance, executará o algoritmo incorretamente, a não ser que a possibilidade de inverter os valores das variáveis seja detalhada no programa. Por outro lado, se **b** for igual a zero, a interpretação do problema, para um ser humano, pode variar conforme o contexto, o que não invalida o algoritmo. Já o computador, salvo instrução explícita em contrário, fará uma divisão por zero ($\mathbf{a} \bmod \mathbf{b}$), e retornará uma mensagem de erro.

Um programa de computador, em resumo, é a expressão de um algoritmo por meio de uma linguagem de programação (linguagem formal), levando em conta as características funcionais do computador onde será executado. Então, ao escrever um programa, uma lógica correta não basta: a adaptação da lógica às características da máquina também é essencial.

¹⁷ “The execution of an algorithm must not normally involve subjective decisions, nor must it call for the use of intuition or creativity. Hence a cooking recipe can be considered to be an algorithm if it describes precisely how to make a certain dish, giving exact quantities to use and detailed instructions for how long to cook it. On the other hand, if it includes vague notions as ‘add salt to taste’ or ‘cook until tender’ then we would no longer call it an algorithm.”

2.2. Aspectos práticos da computação

Em sentido amplo, o termo computador denomina uma máquina capaz de processar ao menos um *modelo informacional*, onde os sinais de entrada e saída adquirem significado, dentro de um contexto definido por seres humanos. Isso implica no estabelecimento de uma correspondência aproximada entre os dados processados pelo computador e as variáveis do problema a ser resolvido. Mais precisamente, podemos dizer que um computador é uma máquina através da qual dados são coletados, armazenados e transformados *com propósitos definidos*. Tal conceito de computador, contudo, envolve problemas muito além do nosso escopo, como por exemplo a distinção entre computadores analógicos e computadores digitais.

A inteligência artificial, objeto de crítica deste trabalho, tem fundamentado suas promessas e realizações nos computadores eletrônicos digitais. Portanto, para os nossos fins, podemos definir o computador como uma máquina capaz de executar programas, que são algoritmos traduzidos para uma linguagem formal, portanto obedecendo a um conjunto estrito de comandos e formas bem-definidas para representação dos dados. No computador eletrônico digital, em particular, os dados e instruções são representados na forma binária, e fisicamente se traduzem em sinais elétricos, associados aos algarismos 0 (zero) e 1 (um). Daqui para a frente, a palavra computador será usada, neste texto, como sinônimo para computador eletrônico digital.

Os computadores, na verdade, só executam programas escritos em linguagem de máquina, que se restringe aos códigos de instruções e formatos de dados que podem ser processados diretamente pela CPU (unidade central de processamento). Se o programa for escrito em outra linguagem, deve antes ser traduzido, para depois ser executado. Esse tipo de tradução é feito automaticamente, por programas especializados, chamados tradutores de linguagens de programação.

Toda linguagem de programação é uma linguagem formal, baseada em uma gramática rígida, em que não existem exceções às regras estabelecidas. Portanto, ao contrário do que acontece na execução de algoritmos por seres humanos, a execução de programas de computador não admite qualquer tipo de ambigüidade. Para a máquina, cada instrução tem um papel rigorosamente definido, e salvo algum erro de processamento, fornecerá exatamente a mesma resposta, quando executada em situações idênticas. O computador, ou mais precisamente, sua unidade central de processamento, não atribui às instruções qualquer significado. A CPU simplesmente processa símbolos binários, de acordo com as regras definidas pela arquitetura da máquina.

Os computadores são máquinas determinísticas. Portanto, um certo conjunto de dados, submetido a uma certa ordem de operações, produzirá sempre os mesmos resultados. Esta característica dos computadores é um dos seus grandes atrativos, e uma das principais razões do seu enorme sucesso. Devido ao determinismo, temos a certeza que um algoritmo que funciona, *funciona sempre*, no contexto em que se aplica¹⁸. Por outro lado, há situações em que um determinismo absoluto é indesejável, como no caso dos algoritmos probabilísticos, em que se faz imprescindível a obtenção de resultados variáveis, mesmo diante de um único conjunto de dados. Isso acontece, por exemplo, quando os computadores são usados em simulações. A fim de satisfazer a esta necessidade, os computadores são dotados de instruções para a geração de números aleatórios. Tal expediente é de grande utilidade prática, mas não anula o determinismo característico dos computadores: as seqüências de operações estabelecidas pelo programa continuam sendo seguidas inflexivelmente, e para um determinado conjunto de dados, um determinado conjunto de instruções fornecerá sempre o mesmo resultado. Os números aleatórios apenas introduzem novos dados para processamento, permitindo flutuações nos resultados obtidos.

A fim de se utilizar um computador na solução de determinado problema, os seguintes passos precisam ser cumpridos com êxito: (1) O problema deve ser modelado, com a conseqüente elaboração de um algoritmo que o resolva de modo satisfatório; (2) O algoritmo deve ser traduzido para a linguagem de máquina do computador a ser utilizado, geralmente após ser escrito em uma linguagem de programação de alto nível¹⁹; e (3) O computador escolhido deve executar o programa, em um intervalo de tempo compatível com a natureza do problema.

Este roteiro, de aparência um tanto simplória, se depara com dificuldades importantes, não só de ordem tecnológica e conceitual, como também de ordem administrativa. Na solução de problemas com o auxílio dos computadores, a gestão de recursos é

¹⁸ De fato, os computadores modernos incorporam vários dispositivos físicos e lógicos destinados especificamente a aumentar, o quanto possível, a previsibilidade da sua operação. O mais conhecido é o “bit de paridade”, que introduz uma redundância na representação dos dados, a fim de reduzir a possibilidade de alteração indesejável das palavras binárias, em função de ruídos eletromagnéticos.

¹⁹ Linguagem de programação de *alto nível* é aquela que possui uma sintaxe mais rica do que a linguagem de máquina, definindo um conjunto mais elaborado de comandos e estruturas de armazenamento de dados. Em tese, qualquer programa poderia ser diretamente escrito em linguagem de máquina, mas na prática, as linguagens de alto nível são essenciais para que os programadores trabalhem com produtividade.

freqüentemente um aspecto da maior importância. Embora o presente texto passe ao largo dessa questão, deve ficar claro, no entanto, que além dos problemas aqui discutidos, os limites e possibilidades do computador também dependem, em larga medida, da organização sócio-econômica necessária para colocá-los em marcha, envolvendo capital, tecnologia e disponibilidade de profissionais qualificados, entre outros fatores.

Feita essa ressalva, cabe considerar que a transformação de um algoritmo em um programa de computador nem sempre é uma operação trivial. O algoritmo, na sua forma *pura*, lida com entidades matemáticas idealizadas (símbolos e grandezas). Para se transformar um algoritmo em programa, tais entidades têm que ser adaptadas às possibilidades de representação da linguagem a ser utilizada. Além disso, a representação tem que se manter fiel aos objetos representados, no decorrer do processamento, que às vezes implica em bilhões ou trilhões de operações, aplicadas sucessivamente a um conjunto de dados. Nos algoritmos numéricos, por exemplo, deve-se ter um cuidado especial para que pequeninos erros de aproximação, acumulados repetidamente, não se transformem em grandes erros, que inviabilizariam a utilização do programa. Tais erros de aproximação, freqüentemente inevitáveis, ocorrem porque cada número real, no interior de um computador, tem que ser representado como uma seqüência finita de bits, e na conversão da base decimal para a base binária, nem sempre é possível encontrar uma representação exata para a parte fracionária dos números (um número absolutamente trivial, como 0,6, ao ser convertido para a base binária transforma-se em uma dízima: 0,100100100...).

Retornando ao item (3), acima, dizíamos que o computador escolhido tem que rodar o programa. Precisamos acrescentar: rodar o programa em um tempo aceitável, e dentro dos limites de espaço para armazenamento de dados determinados pelas características do hardware. Então, não basta que um problema possa ser resolvido por meio de um algoritmo; é necessário que este algoritmo seja executável em uma máquina real. Este ponto é enfatizado por Lewis e Papadimitrou (2000, p.261):

Com os grandes avanços da tecnologia da computação das últimas décadas, seria razoável esperar que todos os problemas do primeiro tipo [os que podem ser resolvidos através de algoritmos] possam agora ser resolvidos de uma maneira satisfatória. Infelizmente, a prática da computação revela que muitos problemas, embora solúveis a princípio, não podem ser resolvidos em qualquer sentido prático por computadores, devido às *excessivas exigências de tempo* (grifo no original).

A insolubilidade dos referidos problemas, por meios algorítmicos, é uma questão tecnológica, mas em um sentido muito especial. Não se trata apenas de esperar por computadores eletrônicos digitais com maior poder de processamento e armazenamento de dados. Quando as exigências de tempo ou espaço de armazenamento para executar um dado algoritmo são exponenciais, em relação ao tamanho da entrada de dados, não podemos esperar processá-lo, usando computadores eletrônicos digitais. O importante é que essa limitação não se refere só aos computadores atuais, mas a qualquer tipo de computador eletrônico digital que possa ser construído, dentro dos referenciais científicos e tecnológicos do presente.

A fim de situar a questão mais concretamente, recorreremos à análise de um problema clássico, conhecido como o “problema do caixeiro viajante”. Suponhamos que um representante de vendas tem que visitar dez cidades, e deseja fazê-lo gastando o mínimo possível com combustível. Conhecendo a distância entre cada cidade do roteiro, e considerando que as cidades podem ser visitadas em qualquer ordem, qual é a seqüência de visitas mais econômica? Para resolver o problema, poderíamos fazer um algoritmo que gerasse todas as seqüências possíveis, verificando a quilometragem total de cada uma. No caso em foco, tal abordagem seria viável porque o número de seqüências, para dez cidades é $9!$, ou 362880, o que é pouco, para um computador pessoal moderno. Porém, na medida em que o número de cidades aumenta, o número de seqüências a serem verificadas cresce abruptamente, em um processo que se costuma chamar de *explosão combinatória*. A tabela 1 ilustra algumas instâncias do problema, considerando a possibilidade de se calcular cerca de dez milhões (10^7) de roteiros por segundo, um valor dentro das possibilidades dos microcomputadores atuais:

| No. de Cidades | No. de Seqüências a Analisar | Tempo aproximado para o cálculo exaustivo (10^7 seqüências / s) |
|----------------|------------------------------|--|
| 10 | 362.880 | $3,6288 \times 10^{-02}$ s |
| 12 | 479.001.600 | 47,9 s |
| 15 | $1,3077 \times 10^{12}$ | 36h19min |
| 20 | $2,4329 \times 10^{18}$ | 7.714 anos |
| 40 | $8,1592 \times 10^{47}$ | $2,5873 \times 10^{33}$ anos |

Tabela 1: Instâncias do problema do caixeiro-viajante

Os valores da tabela falam por si, mas podemos enfatizar os enormes saltos no tempo requerido para calcular os roteiros, na medida em que o número de cidades aumenta. Além disso, a tabela não leva em conta o espaço de armazenamento de dados necessário para manter e comparar os roteiros propostos, que estaria totalmente fora das nossas possibilidades tecnológicas atuais. O problema, portanto, está nos limites da física atual. Algumas leis e constantes fundamentais, tais como a velocidade da luz e o diâmetro de um átomo de hidrogênio, impõem barreiras definidas ao desempenho que podemos esperar obter dos computadores eletrônicos.

A rigor, o algoritmo do caixeiro viajante admite várias otimizações, que reduziriam o número de buscas a realizar. Contudo, não se conhece nenhum recurso capaz de melhorar o *fator de crescimento* do número de buscas, em relação ao número de cidades. Portanto, tratam-se de recursos paliativos: podemos aumentar um pouco o número de cidades a ser tratado, mas qualquer grande aumento provocaria, novamente, uma explosão combinatória.

O que está em jogo é a *relação entre o tamanho da entrada de dados e o tempo que o algoritmo leva para rodar*. Este tempo é proporcional ao número de operações elementares a serem executadas pelo computador, tais como somas e subtrações²⁰. No caso do problema do caixeiro viajante, o número de operações cresce com o fatorial do número de cidades, o que é um índice de crescimento intolerável, em termos práticos. Portanto, embora o algoritmo para a solução do problema seja simples, ele só se aplica para um número de cidades bastante limitado.

²⁰O conceito de operação elementar, é detalhado na próxima seção.

O estudo do desempenho dos algoritmos, em condições extremas, pertence a um campo chamado complexidade computacional. Este campo da computação tem como finalidade precípua nos ajudar a identificar os algoritmos que têm interesse prático na solução de problemas, no que diz respeito ao tempo de processamento. Normalmente, considera-se que os algoritmos nos quais a taxa de crescimento do número de operações é uma função polinomial do tamanho da entrada de dados, são algoritmos viáveis. Por outro lado, os algoritmos nos quais o número de operações cresce de modo exponencial, com o crescimento da entrada de dados, são geralmente considerados inviáveis, a não ser para entradas de dados de tamanho reduzido.

No próximo capítulo, “Jogos e Computadores”, retomaremos o problema da complexidade computacional, e veremos como ele tem um impacto decisivo no tratamento computacional dos jogos. Desde já, vale salientar que grande parte dos esforços despendidos no aprimoramento dos programas de xadrez tem como objetivo superar a barreira imposta pela natureza exponencial dos algoritmos de cálculo de variantes.

2.3. As máquinas e seu desempenho

Embora o nome computador geralmente esteja associado a um objeto material, na realidade todo computador é um dispositivo lógico com um suporte físico – uma combinação de hardware e software. O hardware é o conjunto de circuitos eletrônicos, mais os dispositivos periféricos (mouse, monitor, etc.) que possibilitam a interação entre o usuário e a máquina. O software é o conjunto de dados e instruções, logicamente estruturados, que se exprimem fisicamente através dos sinais elétricos no interior dos circuitos²¹. Sem algum tipo de software instalado, um computador simplesmente não funciona, mas, sem um hardware que lhe dê suporte, o software é uma criação inacabada. Combinados, hardware e software dão ao computador sua reconhecida capacidade no auxílio à solução de problemas.

Há quatro elementos necessários e suficientes para caracterizar um computador: (1) uma ou mais unidades de processamento, capazes de executar um conjunto de instruções para leitura, processamento e escrita de dados²²; (2) um espaço para armazenamento de

²¹ O caráter um tanto abstrato do software é admiravelmente expresso pelo termo francês “logiciel”.

²² Compreendendo-se leitura como a inserção de um conjunto de dados no interior do espaço lógico da máquina, e escrita como o processo inverso, de apresentação de um conjunto de dados para o espaço lógico externo à máquina.

dados e programas; (3) um ou mais dispositivos de entrada e saída de dados; (4) um mecanismo de controle, que coordena a relação entre os elementos (1), (2) e (3). A complexidade desses elementos, assim como a forma como estão combinados, variam conforme as características do computador, mas, em essência, eles permanecem sempre os mesmos.

Dos mais simples aos mais poderosos, tudo o que os computadores fazem é executar operações aritméticas e operações da lógica booleana, sobre dados representados por cadeias de 0s (zeros) e 1s (uns). Logo, o desempenho diferenciado dos diversos tipos de computador depende basicamente de cinco fatores:

- (a) A velocidade bruta de processamento, isto é, o número de instruções que podem ser processadas em um intervalo de tempo padrão. Em geral, esta velocidade é medida em MIPS – milhões de instruções por segundo, ou MFLOPS – milhões de operações de ponto flutuante por segundo, e depende da frequência de chaveamento (clock) dos circuitos, medida em Hertz (ciclos por segundo).
- (b) A capacidade de armazenamento de dados do hardware, medida em bytes (palavras binárias, com oito bits cada). Os computadores atuais têm de três a quatro níveis de armazenamento, um estruturado sobre discos magnéticos e os demais, sobre circuitos semicondutores.
- (c) As características das instruções elementares que a máquina executa. Se uma instrução não está disponível diretamente no hardware, ela tem que ser implementada via software, o que geralmente aumenta bastante o tempo de processamento²³.
- (d) A qualidade do sistema operacional (SO), que é o nome dado ao software que controla as operações do hardware. O desempenho do SO é extremamente importante, sobretudo considerando que os computadores modernos quase sempre executam diversas tarefas concorrentemente.
- (e) A otimização do software aplicativo, utilizado para executar as tarefas de interesse do usuário.

A base fundamental para o projeto dos computadores modernos é a máquina de von Neumann (Tanenbaum, 1992, p.13; Akl, 1989, p.xi), que difundiu o conceito de programa armazenado. Implementado no computador EDSAC, no início da década de

²³ Este aspecto é mais crítico no que se refere à aritmética de alto desempenho, que é fundamental para a computação científica e para o processamento de gráficos em alta resolução, e/ou em tempo real. Até cerca de quinze anos atrás, os microprocessadores típicos não possuíam instruções para trabalhar com números de ponto flutuante, mas apenas com números inteiros. A aritmética de ponto flutuante, implementada em software, era extremamente lenta, para os padrões atuais.

1950, o modelo de von Neumann foi revolucionário, em termos de flexibilidade. Ao permitir que as instruções a serem executadas compartilhem com os dados o mesmo espaço de armazenamento, a arquitetura von Neumann eliminou os tediosos procedimentos de programação anteriores, baseados na operação manual de chaves e conectores. Em linhas gerais, execução das instruções em uma máquina von Neumann dá-se de forma rigorosamente seqüencial: a unidade central de processamento (CPU) busca uma instrução na memória, juntamente com os seus operandos, executa a operação requerida e envia o resultado novamente para a memória, repetindo esse ciclo, até que o programa termine.

Ao proporcionar um modelo conceptual padrão para o projeto de computadores de uso geral, a arquitetura von Neumann inaugurou uma nova etapa na história da computação. Entretanto, a história da computação é também a de uma busca incessante pelo processamento cada vez mais rápido de grandes volumes de dados e, com o tempo, a arquitetura von Neumann, na sua forma elementar, demonstrou-se insuficiente para se beneficiar da miniaturização eletrônica, e para atender ao aumento da demanda por programas de alto desempenho. Conseqüentemente, surgiram arquiteturas de computador alternativas, visando obter o máximo da tecnologia de semicondutores.

Todos os computadores trabalham baseados em um *clock*, ou relógio, que dá o compasso para o funcionamento dos circuitos digitais. A cada ciclo do relógio, o chaveamento dos circuitos realiza uma operação, ou parte de uma operação elementar. As arquiteturas de computador alternativas à de von Neumann têm em comum a tentativa de executar mais operações em um menor número de ciclos de relógio, introduzindo algum tipo de paralelismo.

Assim, por exemplo, as *máquinas vetoriais*, ao invés de operar com números aos pares, são capazes de realizar operações aritméticas sobre pares de vetores. As CPUs do tipo *pipeline*, por sua vez, assemelham-se a linhas de montagem, pois executam, simultaneamente, fragmentos de instruções sucessivas, de forma que, quando uma instrução termina de ser executada, outras já estão em processo de execução. (Tanenbaum, 1992, p.27-29). Hoje em dia, são bastante comuns os computadores com várias CPUs trabalhando em paralelo, executando porções de um mesmo programa ou programas distintos. Quando se estabelece uma cooperação entre diferentes máquinas, conectadas via rede, fala-se em PVMs – máquinas paralelas virtuais.

Outra técnica de paralelismo corriqueira é o uso de *memórias cache*, que são circuitos de armazenamento de altíssimo desempenho, fisicamente próximos à CPU²⁴, e conectados diretamente a ela por uma via de dados exclusiva. Constantemente, a CPU traz para a memória cache os dados e instruções que provavelmente serão necessários no decorrer do processamento, de acordo com critérios estabelecidos pelos projetistas do hardware. Se os dados disponíveis em cache se tornarem realmente necessários, como quase sempre ocorre, poupa-se o tempo que seria despendido no acesso até a memória principal. Como o *caching* é feito ao mesmo tempo em que a CPU executa instruções, ele acelera sobremaneira a execução dos programas, ao impedir que a CPU fique ociosa, enquanto os dados e instruções são buscados na memória principal.

Com uso de técnicas paralelas, pode-se obter ganhos de performance lineares ou, em casos muito especiais, como o cálculo com vetores, ganhos polinomiais (digamos, faz-se num tempo t um produto de vetores que antes requeria um tempo t^2). Tais ganhos são importantes em termos práticos, e têm proporcionado grandes avanços na solução de problemas difíceis, como a previsão do tempo e a simulação aerodinâmica. Entretanto, os algoritmos de natureza exponencial continuam sendo intratáveis, porque qualquer tentativa de aumentar o paralelismo de forma exponencial logo esbarraria em limitações físicas evidentes. O que as arquiteturas paralelas têm viabilizado é a execução de algoritmos exponenciais para conjuntos de dados um pouco maiores do que o permitido pelas máquinas convencionais. Para certos tipos de problemas, como o cálculo de variantes no xadrez, mesmo esses ganhos modestos são muito significativos, e não surpreende que o Deep Blue fosse uma máquina maciçamente paralela, com quase mais de 400 processadores trabalhando em conjunto.

²⁴ Quando os circuitos operam a centenas ou milhares de MHz, a proximidade física dos componente é importante, para garantir o perfeito sincronismo, sem retardo dos sinais elétricos. Um microcomputador comum, atualmente, trabalha a mais de 1 GHz, ou bilhão de ciclos por segundo. No intervalo de um ciclo, ou seja, um bilionésimo de segundo, a luz se move aproximadamente 30cm no vácuo, e um sinal elétrico, em um circuito, menos do que isso.

3. Jogos e Computadores

3.1. Terminologia

Em linhas gerais, os jogos podem ser divididos em dois grandes grupos: aqueles cuja lógica interna independe do mundo material, e aqueles essencialmente ligados ao mundo material. Neste último grupo, estão futebol, tênis, hóquei, beisebol, basquete e qualquer outro jogo, individual ou por equipes, cuja prática está necessariamente associado às leis físicas. No primeiro grupo, estão os jogos de cartas e jogos de tabuleiro, cuja representação por meio de objetos físicos é contingente, mas não necessária, do ponto de vista lógico. A estes daremos o nome de jogos abstratos.

Em um jogo abstrato, o suporte físico usado pelos jogadores condiciona-se apenas aos hábitos culturais, não à essência do jogo. Assim, no jogo de damas, tanto faz se as peças são representadas por cilindros de madeira, pedras coloridas, conchas, grãos de arroz e feijão, etc. O tabuleiro pode ser de papel, metal, vidro, madeira, desenhado na areia, ou inexistente, caso os jogadores (e o juiz) sejam capazes de visualizar o jogo sem qualquer suporte físico exterior ao cérebro. Importa apenas que os objetos escolhidos, ou os esquemas mentais dos jogadores representem com exatidão cada peça presente no jogo, e as relações entre elas. Da mesma maneira, nos jogos que usam baralho, os desenhos das cartas podem ser alterados arbitrariamente, desde que se preservem as convenções de ordem e as quantidades de cada tipo de carta. No xadrez, preservadas as relações lógicas entre as peças e o tabuleiro, fatores como as cores das casas, o design e o tamanho das peças são irrelevantes²⁵.

Nos jogos intrinsecamente ligados ao mundo físico, a situação é outra: as regras especificam itens como o peso da bola, o tamanho do campo e as dimensões das traves, essenciais para a caracterização de cada jogo. No caso do futebol, por exemplo, variações no peso e tamanho da bola, nas dimensões das traves, no tamanho do campo e

²⁵ Isto é, irrelevantes para a lógica do jogo, embora, provavelmente, relevantes para o desempenho dos jogadores. No Clube Baiano de Xadrez, conheci um tabuleiro de xadrez destinado a um jogador cego, em que as casas pretas foram recobertas por lixas, e existiam pinos onde as peças podiam ser encaixadas. A deficiência visual do nosso colega não o impedia de jogar xadrez, mas impunha uma representação especial do tabuleiro e das peças, a fim de que ele pudesse jogar com segurança. Vale notar que mesmo os jogadores com visão perfeita têm preferências nítidas por certos modelos de peças, e frequentemente, a utilização de peças de modelos não habituais se reflete decisivamente no desempenho dos jogadores.

no tipo de piso determinam a existência de duas modalidades desportivas completamente diferentes, o futebol de campo e o futebol de salão.

É perfeitamente sensato jogar qualquer jogo abstrato “de cabeça”, ou com a ajuda de papel e lápis, bastando apenas que os participantes consigam fazê-lo. Nesses jogos, as regras se aplicam em um plano estritamente formal, tornando praticamente impossíveis as interpretações divergentes. Os jogos dependentes do mundo físico, ao contrário, só podem ser jogados através de movimentos característicos dos atletas. Com isso, a aplicação das regras está sujeita à interpretação dos árbitros, eventualmente apoiados por dispositivos eletrônicos, como filmadoras e máquinas fotográficas de alta velocidade.

Os chamados jogos de azar, como a roleta, o bingo e os dados, considerados no seu caráter estatístico, independente de eventuais imperfeições físicas (ou “vícios”) dos equipamentos, não são jogos de verdade, sob uma perspectiva computacional. Nos dados e no bingo, não há nada que se possa fazer para influenciar o resultado das partidas, e na roleta, tudo o que se pode esperar é um algoritmo para minimizar as perdas do jogador, já que, no longo prazo, a banca sempre vence. O problema dos jogos de azar não é a presença de fatores aleatórios, que também aparecem no gamão, no Banco Imobiliário e no pôquer, por exemplo. O problema é que nos dados e na roleta só há aleatoriedade, de modo que os jogadores simplesmente não exercitam habilidade alguma – ressaltando, mais uma vez, que esta análise se restringe a situações ideais, em que não há vícios nem trapaça.

De agora em diante, nesta dissertação, a palavra jogo será usada para designar os jogos abstratos – aqueles cuja essência se esgota nas suas necessidades lógicas e que, portanto, são prontamente passíveis de um tratamento matemático.

3.2. Características dos jogos

Todos os jogos começam a partir de um arranjo definido das peças (ou cartas). A este arranjo, dá-se o nome de *estado inicial*. Com o desenrolar do jogo, os arranjos das peças mudam, e novos *estados* se sucedem. Um *estado terminal* é um arranjo de peças que define o resultado da partida, caracterizando um empate ou a vitória de um dos competidores. Do ponto de vista do agente, *estado final*²⁶ é um estado terminal

²⁶ O termo usual na IA é “goal state”, isto é, o estado que se almeja.

desejável – dentre os estados terminais possíveis, um dos que seja favorável ao agente, e desfavorável ao seu adversário.

Tomando como exemplo o jogo-da-velha, um dos mais simples jogos abstratos, os elementos fundamentais dos jogos ficam assim caracterizados: (1) O estado inicial caracteriza-se pelo tabuleiro vazio, representado por uma matriz 3x3; (2) Na medida em que cada jogador marca uma posição do tabuleiro, desenham-se diferentes relações entre as marcas ('Xs' e 'Os'), que determinam sucessivos estados; (3) O jogo atinge um estado terminal quando um dos jogadores alinha três marcas, na horizontal, vertical, ou diagonal, ou ainda quando o tabuleiro fica completamente preenchido, sem que nenhum dos jogadores tenha alinhado três marcas. Cada marca realizada por um jogador no tabuleiro equivale a um operador, ou seja, uma ação que determina a transição entre dois estados sucessivos.

Formalmente, um jogo caracteriza-se por:

- (a) Um espaço de estados, que é o conjunto de todos os arranjos possíveis das peças, respeitadas as regras do jogo, isto é, todos os arranjos de peças que podem ser atingidos, a partir do início do jogo, através de uma sucessão de jogadas legais.
- (b) Um estado inicial, que pode ser sempre o mesmo, ou um, dentre vários estados iniciais possíveis. Em outras palavras, há um conjunto de estados iniciais, que pode ser um conjunto unitário. No xadrez e nas damas, o estado inicial (arrumação das peças) é sempre o mesmo, mas no dominó o estado inicial de cada rodada (mãos dos jogadores) é aleatoriamente definido, sendo uma dentre todas as combinações possíveis de peças;
- (c) Um conjunto de operadores (jogadas), que determinam a passagem entre estados sucessivos. A cada estado do jogo, exceto nos estados terminais, deve haver pelo menos um operador disponível, para que o jogo tenha continuidade. O leque de operadores disponíveis é dado pelas regras do jogo. No xadrez, ambos os jogadores dispõem, no primeiro lance, de vinte jogadas legais, número que se altera na medida em que a partida se desenrola;
- (d) Um conjunto de estados terminais, que caracterizam um empate ou a vitória de um dos jogadores. No jogo-da-velha, a definição de estado final independe da história pregressa do jogo, ao contrário do que ocorre nas damas ou no xadrez. No xadrez, uma mesma posição pode ou não caracterizar um estado final, como nos casos em que o correm repetições de lances, conforme detalhado a seguir.

Em alguns jogos, o estado terminal está associado a uma *função-recompensa*, ou *função-utilidade* (Russel e Norvig, 1995, p.124), que estabelece um valor numérico para o resultado do jogo. No xadrez, o resultado é apenas empate, vitória ou derrota²⁷. No jogo de dominó, porém, deve-se somar os valores das peças nas mãos dos jogadores, e serão mais penalizados aqueles que tiverem em mãos um maior número de pontos quando o jogo terminar. A característica da função-recompensa associada a um jogo é fator preponderante na formulação de estratégias. Se no xadrez o que importa é dar xeque-mate, isso justifica o “sacrifício” eventual de todas as peças desnecessárias para se chegar ao objetivo. No jogo de buraco, por outro lado, não importa apenas “bater”, mas também livrar-se o mais rapidamente das cartas de maior valor, para o caso de um dos adversários “bater” primeiro, uma vez que as cartas que permanecerem na mão de cada jogador contarão pontos negativos.

No xadrez e nas damas, assim como em outros jogos, os adversários podem empatar de comum acordo, o que não significa que foi alcançado um estado terminal. No xadrez, em particular, se uma mesma posição das peças se repetir por três vezes, os jogadores *podem* requerer um empate, mas não são obrigados a fazê-lo, o que significa que a caracterização de um estado terminal, nessas circunstâncias, depende da história pregressa do jogo e da vontade de um dos adversários. No caso, se um dos jogadores requerer o empate, o outro será obrigado a aceitá-lo, de modo que um estado terminal fica de fato caracterizado, diferentemente do que ocorre nos empates por comum acordo. Esse detalhe tem conseqüências importantes para o xadrez computacional, porque os programas precisam estar preparados para responder a propostas de empate, além de reconhecer a repetição de posições, sobretudo como um recurso para empatar em posições inferiores.

Em jogos como as damas, o xadrez e o Go, a transição de um estado a outro – ou seja, cada jogada realizada – depende apenas da deliberação dos jogadores. Como esses jogos não envolvem o elemento “sorte”, na passagem de um estado a outro, nós os chamaremos de jogos não-probabilísticos. Já no gamão, além das decisões dos jogadores, a sorte influencia na definição das jogadas, através dos dados. Chamaremos de jogos probabilísticos aqueles em que a transição entre dois estados depende não somente de uma deliberação do jogador a quem cabe jogar, mas também do fator

²⁷ Ao contrário do que pensam alguns principiantes, que contam as peças do tabuleiro, ou somam seus hipotéticos valores numéricos, para saber quem ganhou o jogo. De fato, um dos importantes aspectos estéticos do xadrez é a possibilidade de se vencer por meios “econômicos”, com poucas peças, coordenadas para dar xeque-mate ao adversário, em situações em que o lado vencedor tem ampla desvantagem material (ou seja, menos peças e/ou peças de menor valor absoluto).

“sorte”, introduzido por algum mecanismo de funcionamento aleatório (em geral, rolagem de dados, como no gamão, ou distribuição de cartas embaralhadas, como no pôquer).

Uma segunda classificação, igualmente importante, diz respeito aos elementos de que os jogadores dispõem para tomar suas decisões. Nesse aspecto, diz-se que os jogos podem ser de informação perfeita ou de informação imperfeita. Um jogo de informação perfeita é aquele em que todos os jogadores têm acesso irrestrito à disposição das peças no campo de jogo, como ocorre nas damas e do xadrez, onde o tabuleiro é aberto, e não existe qualquer arranjo de peças oculto. Em um jogo de informação imperfeita, por outro lado, os jogadores só têm acesso parcial ao estado do jogo, em cada instante. Em geral, isso ocorre em todos os jogos de cartas, e também nas diversas modalidades de dominó, onde uma parte das peças (cartas) permanece oculta para cada jogador.

As duas classificações discutidas acima não se excluem mutuamente, podendo se combinar de quatro formas, conforme a tabela abaixo:

| | Não-Probabilísticos | Probabilísticos |
|--|---|---|
| Informação Perfeita | Xadrez Go Damas Jogo-da-Velha Dominó ⁽¹⁾ | Gamão Banco Imobiliário |
| Informação Imperfeita | Dominó ⁽²⁾ Copas | Pôquer Buraco Dominó ⁽³⁾ |
| <p>⁽¹⁾ Duas pessoas, iniciando com quatorze peças cada. ⁽²⁾ Em duplas. ⁽³⁾ Duas pessoas, iniciando com sete peças cada.</p> | | |

Tabela 2: Alguns jogos clássicos

Como se vê, o dominó pode ser classificado de três formas diferentes, a depender da modalidade de jogo. Se há dois adversários iniciando com quatorze peças cada, basta que cada jogador conte as próprias peças e as peças da mesa, para saber quais são as peças do adversário. Assim, nessa modalidade de dominó, tem-se um jogo de informação perfeita, e como não há peças de cabeça para baixo, para serem apanhadas aleatoriamente, tem-se também um jogo não-probabilístico. No dominó por duplas, não há peças na mesa, de cabeça para baixo. Mas como as peças estão distribuídas por

quatro participantes, nenhum deles tem acesso a combinação exata de peças nas mãos de cada um dos outros²⁸. Tem-se, portanto, um jogo de informação imperfeita, mas não-probabilístico, porque cada jogador, quando lhe cabe jogar, decide o que fazer sem qualquer intervenção da “sorte” ou do “azar”. Finalmente, se dois adversários jogam um contra o outro, começando com sete peças cada, inicialmente nenhum dos dois pode ter certeza sobre que peças estão nas mãos do oponente e que peças estão no “monte”. Além disso, quando se faz necessário pegar uma peça no monte, para completar a jogada, a escolha da peça a ser apanhada é aleatória, e influencia o desenrolar do jogo. Portanto, tal modalidade de dominó é um jogo probabilístico de informação imperfeita.

Pode parecer estranha a classificação do jogo de copas e de uma modalidade de dominó como jogos não probabilísticos, pois em ambos os casos, o embaralhamento das peças (cartas), antes do início da partida, influencia as possibilidades e estratégias dos jogadores. Porém, a influência do fator “sorte” cessa aí, isto é, não são inseridos novos elementos de aleatoriedade no decorrer do jogo. Nos jogos que chamamos de probabilísticos, ao contrário, cada jogada é precedida pela introdução de um elemento de aleatoriedade – tipicamente, o lançamento de dados, como no gamão e no banco imobiliário.

Por fim, vale à pena uma distinção entre jogos e quebra-cabeças: nos jogos, há dois ou mais participantes interessados em otimizar seus resultados, em detrimento do resultado do adversário (ou adversários). Nos quebra-cabeças, há um desafio a ser ultrapassado, mas não há adversários a serem enfrentados. Assim, ainda que se utilize algum critério de desempenho, como o tempo necessário para a conclusão do quebra-cabeça, não existem um ou mais adversários lutando para atrapalhar o “solucionador”²⁹. Reconhecida esta diferença fundamental, deve-se admitir que os quebra-cabeças também são um bom campo de provas para a inteligência artificial. Na prática, um quebra-cabeça qualquer pode se transformar em objeto de competição – por exemplo, cronometrando-se o tempo necessário para resolvê-lo.

Os quebra-cabeças são reversíveis, isto é, se uma determinada ação não for bem sucedida, o resolvidor poderá revertê-la, retornando a um estado anterior, para seguir outros caminhos (supondo que se lembre de como fazê-lo). Nos jogos, porém, a

²⁸ Embora, na medida em que o jogo avança, essa composição possa ser inferida, primeiro de forma grosseira, e posteriormente, com alto grau de certeza, por meio das jogadas realizadas e da contagem das peças.

²⁹ “Solucionador”, em oposição a jogador, já que este último enfrenta um ou mais adversários, e o outro, não.

possibilidade de retornar a um estado, quando existe, depende da cooperação do adversário. No xadrez, é comum a repetição de posições, quando isso convém a ambos os jogadores, mas, em outros jogos, como as damas, as regras para a movimentação das peças fazem com que o retorno a posições passadas seja impossível, na maior parte das vezes. Por isso, seguir um caminho errado, em um quebra-cabeça, implica em gastar mais tempo na solução, enquanto seguir um caminho errado, em um jogo, pode significar um comprometimento irreversível do resultado da partida. Obviamente, pode-se introduzir, na solução desse ou daquele quebra-cabeça, uma regra que proíba o retorno a estados anteriores. Isso seria, contudo, apenas uma opção para dificultar a solução. Nos jogos, a eventual impossibilidade de retornar a um estado anterior não é uma opção, mas uma característica essencial.

O enfrentamento direto entre adversários dá aos jogos um apelo lúdico e emocional mais forte, e também permite colocar à prova a força dos programas de forma mais direta e atraente do que os quebra-cabeças. O segundo match entre Kasparov e o Deep Blue e Kasparov provocou uma das maiores concentrações de acesso a um site Web, em todos os tempos; naturalmente, não se poderia esperar que um computador *solitário*, resolvendo um quebra-cabeças, despertasse o mesmo interesse por parte do grande público – mesmo que fosse um quebra-cabeças que nenhum ser humano jamais resolveu. Nem tampouco seria razoável esperar o mesmo sucesso de um match disputado entre dois computadores, por mais fortes que fossem no xadrez. Mas, independente do fator *relações públicas*³⁰, os jogos clássicos são muito especiais para a computação, porque o desempenho humano nesses jogos vem sendo aprimorado no decorrer do tempo, porque há pessoas altamente qualificadas em computação, dispostas a contribuir com o seu conhecimento para o desenvolvimento de programas cada vez melhores, assim como jogadores de elite, dispostos a testar o seu desempenho diante dos programas.

3.3. Os jogos como problemas de busca

Conforme discutido na seção 3.1, todo jogo se caracteriza por uma sucessão de estados, onde a escolha dos operadores (jogadas) visa um objetivo bem definido (um estado terminal favorável, ou *estado final*). Jogar bem, então, significa escolher bem cada jogada. Para tanto, é importante que o jogador preveja, com a máxima precisão possível,

³⁰ Termo usado pelo Sr. Matthew Wisnioski, mestrando de Princeton, em sua palestra “Playing Games: Chess, Automata and Artificial Intelligence”, por mim assistida no History of Science Society 2001 Annual Meeting.

as alternativas do adversário e a suas próprias, para ter segurança sobre as conseqüências de suas escolhas³¹. No xadrez, dá-se o nome de *cálculo de variantes* a essa tentativa de visualizar as vias prováveis para o desenrolar do jogo, a partir de uma determinada posição. Essencialmente, calcular variantes significa explorar as possibilidades de um espaço lógico, e é nesse sentido que a Inteligência Artificial compreende os jogos como problemas de busca³².

Tecnicamente, problema de busca é aquele cuja solução requer, por parte do *agente*³³ (aqui, o jogador ou resolvidor), a definição de uma seqüência de operadores, a fim de passar de um estado para outro, mais favorável. Tal definição depende da *exploração do espaço de estados*, ou seja, uma análise antecipada dos estados que podem ser atingidos, e por que caminhos, para se determinar o melhor operador em uma dada situação – aquele que mais aproxima o agente do seu objetivo.

Por exemplo, ao sair da minha residência para o Instituto de Física da UFBA, tenho que escolher um dentre vários caminhos diferentes. Meu estado inicial é minha residência, e o estado final³⁴ é o Instituto de Física. Posso, então, considerar como estados do meu trajeto cada ponto em que eu seja obrigado a tomar uma decisão sobre o caminho a seguir (esquinas, bifurcações, etc.). O meu espaço de estados fica assim definido como o conjunto de todos esses pontos. Para planejar o meu trajeto, antes de sair de casa, eu poderia tomar em mãos o mapa da cidade e analisar cada um dos percursos possíveis, escolhendo, por fim, o mais satisfatório. O elemento simplificador, nesse exemplo, é o fato de eu saber onde fica o Instituto de Física, e saber que, salvo em circunstâncias extraordinárias (enchentes, acidentes, etc.), é sempre possível chegar lá. Além disso, o mapa da cidade me permite traçar antecipadamente o meu roteiro exato. É mais ou menos como em um quebra-cabeça, no qual as peças são encaixadas para compor a imagem impressa na tampa da caixa.

³¹ É nesse sentido que se diz, às vezes, que o enredo de um filme de espionagem se assemelha a um “jogo de xadrez”, onde as ações são meticulosamente planejadas, levando em consideração as possíveis reações do inimigo.

³² Isto é, busca no sentido de exploração, com a finalidade de se encontrar caminhos favoráveis.

³³ Russell e Norvig (1995) utilizam o termo agente para designar qualquer entidade, idealizada ou real, dotada de alguma autonomia, e capaz de lidar satisfatoriamente com situações complexas, em mundos abstratos ou no mundo físico. Nesse sentido, podemos dizer que o programa que joga algum jogo ou resolve quebra-cabeças é um agente. O termo agente, aqui utilizado, tem uma clara herança cibernética, pois pressupõe a otimização de decisões, diante de circunstâncias externas que variam com o tempo.

³⁴ Lembrando que estado final, aqui, tem o sentido de estado almejado (finalidade).

Nos jogos, a situação se complica, porque sabe-se que existem estados finais desejáveis, mas o agente não pode determinar, antecipadamente, que estado final será alcançado, porque isso depende das decisões do adversário, no decorrer do jogo. Além disso, mesmo em um jogo de informação perfeita, como o xadrez, o número de jogadas possíveis em cada posição faz com que o espaço de estados seja por demais amplo, e o número de caminhos possíveis demasiadamente grande, para permitir uma busca exaustiva, ou seja, a consideração de todos os caminhos possíveis, até o fim do jogo. Estima-se que no xadrez há cerca de 10^{120} partidas válidas (Hsu et al., 1990, p.20), para um total de aproximadamente 10^{43} posições legais (Bonsdorff, Fabel e Riihimaa, 1974, p.12). Isso porque, através das transposições de jogadas, cada posição pode ser alcançada por inúmeros caminhos distintos.

O xadrez, o go, o jogo de damas e todos os demais jogos clássicos têm em comum um número muito grande de partidas possíveis, de modo que as possibilidades criativas dos jogadores humanos não se esgotam – e é por isso tais jogos são interessantes. O jogo-da-velha, ao contrário, geralmente não interessa aos adultos, porque oferece um reduzido número de possibilidades, tornando-se monótono. O número mínimo de jogadas em uma partida de jogo-da-velha é de cinco, o número máximo, de apenas nove. Na medida em que o jogo avança, o número de lances disponíveis vai se reduzindo, até que, se o jogo não tiver terminado antes, na nona jogada resta apenas um lance válido. Como, nesse caso, não há escolha a fazer, pode-se dizer que, em termos práticos, a partida de jogo-da-velha tem no máximo oito jogadas, ao invés de nove.

O jogo-da-velha também tem um reduzido número de posições possíveis; em outras palavras, o seu espaço de estados é restrito. Diante de tal simplicidade, fica fácil determinar, para qualquer posição do jogo-da-velha, qual é a melhor jogada a ser feita. A partir dessa constatação, Rich e Knight (1991, p.9-10) sugerem um algoritmo interessante para jogar o jogo-da-velha perfeitamente. Seria criado um vetor, com tantas posições quantas sejam as posições possíveis do jogo. Depois, cada posição do vetor seria inicializada com um número de um a nove, correspondente ao lance a ser realizado na posição de jogo associada à posição do vetor. O endereçamento do vetor seria feito com base em um sistema de codificação em que cada endereço numérico corresponderia a um estado, ou seja, a uma dada combinação de marcas na matriz do jogo.

Para jogar, a máquina simplesmente traduziria a posição atual do jogo para o endereço correspondente no vetor, e lá encontraria o lance a ser feito. Informalmente, é como se o programa tivesse o jogo-da-velha todo “decorado” e, portanto, seria imbatível. Rich e Knight (op.cit.) sugerem a criação de um vetor com 19.683 posições, correspondentes a

3^9 possíveis combinações de marcas (“X”, “O”, ou espaço em branco). Contudo, considerando que as marcas são sempre feitas alternadamente, o número de posições possíveis no jogo-da-velha é, na verdade, bem menor, ficando abaixo de 6.000³⁵. Ora, uma busca em um vetor de 6.000 posições é feita por um microcomputador moderno em uma fração de milissegundo, de forma que o algoritmo proposto, além de extremamente simples, seria extremamente rápido e econômico, em termos de espaço de armazenamento, para os padrões atuais.

Por exemplo, suponhamos que as casas do tabuleiro do jogo-da-velha fossem numeradas de um a nove, e que atribuíssemos o valor 0 (zero) para um espaço vazio, 1 (um) para uma marca “X” e 2 (dois) para uma marca “O”. Então, seguindo a proposta de Rich e Knight, uma determinada posição do jogo seria representada por um número na base 3, com nove algarismos, onde cada algarismo é determinado pelas posições das marcas. Usando esse sistema, a posição representada na Fig.1 estaria associada ao valor numérico 200011000 na base 3, ou 13.230 na base decimal. Então, a jogada correta para o jogador que tem a vez (no caso, quem marca “O”) estaria na posição de número 13.230 do vetor de jogadas, que conteria o número 4, que é a casa onde o jogador deve marcar “O”.

| | | |
|---|---|---|
| O | | |
| | X | X |
| | | |

Figura 1: Exemplo de posição do jogo-da-velha

A abordagem descrita acima, contudo, só funciona porque o espaço de estados do jogo-da-velha é muito reduzido e o ser humano que faz o programa é capaz de definir o lance ideal para cada situação. Em jogos interessantes, porém, o problema consiste justamente em se encontrar bons lances, diante do enorme número de escolhas à disposição de cada jogador. Porém, nenhum dispositivo construído de acordo com as leis conhecidas da física seria capaz de codificar cada uma das 2×10^{43} posições válidas do xadrez, ou as

³⁵ Uma estimativa aproximada, levando em consideração o número de combinações possíveis de uma, duas, três, quatro, cinco, seis, sete e oito marcas no tabuleiro de jogo da velha. Para uma marca, teríamos ${}^9C_1=9$ possibilidades, para duas, ${}^9C_1 \times {}^8C_1=72$ possibilidades, para três marcas, ${}^9C_2 \times {}^7C_1=252$ possibilidades, para quatro, ${}^9C_2 \times {}^7C_2=756$ possibilidades, para cinco marcas, ${}^9C_3 \times {}^6C_2=1260$, e assim sucessivamente, totalizando aproximadamente 6.000 combinações para até oito marcas, incluídas algumas combinações ilegais, com seis marcas ou mais, pelo fato de ambas as partes apresentarem seqüências de três Os e Xs.

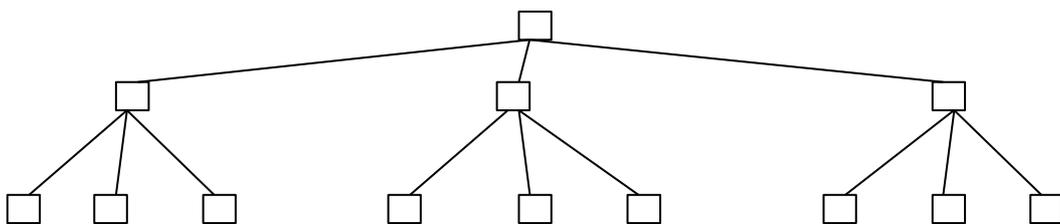
10^{170} posições válidas do Go: simplesmente não há partículas subatômicas suficientes no universo para se construir uma máquina com essa capacidade. Ainda que esta codificação fosse possível, quem (ou *o que*) seria capaz de determinar *o melhor lance* para cada posição? Diante desse impasse, faz-se necessário buscar outros métodos para implementar programas de jogos.

Especificamente, o problema que desafia os estudiosos da computação aplicada aos jogos é o de explorar eficazmente, e dentro dos limites postos pela tecnologia dos computadores eletrônicos digitais, o espaço de estados do jogo em questão, visando otimizar a escolha das jogadas. As seções 3.4 e 3.5, a seguir, tratam desse problema.

3.4. Os problemas da busca exaustiva

Os algoritmos de busca, propriamente ditos, não estabelecem *a priori* os lances a serem feitos para cada posição de um jogo. Ao invés disso, eles *caminham* pelas posições possíveis, procurando prever as seqüências de lances do próprio agente e as respostas adversárias, com a maior antecipação possível.

Graficamente, a estrutura de uma busca costuma ser comparada a uma árvore (geralmente, de cabeça para baixo), onde a raiz é o ponto de partida, e os caminhos alternativos de busca constituem as sucessivas derivações em galhos cada vez mais numerosos. Cada nó de derivação equivale a uma posição intermediária do jogo, entre o estado inicial da busca, os estados que podem ser alcançados em determinado nível de profundidade (já que a busca não prossegue necessariamente até estados terminais). Na literatura técnica³⁶, o número médio de derivações a partir de cada nó é comumente chamado de *branching factor*, expressão que traduziremos como *fator de derivação*. A figura abaixo exemplifica uma árvore de busca com fator de derivação 3, até uma profundidade de duas meias-jogadas³⁷, para um total de 3^2 , ou 9 caminhos possíveis.



³⁶ Por exemplo, Rich e Knight (1991); Russel e Norvig (1995).

³⁷ O termo original é half-move (plural half-moves), ou ply (plural ply). Preferi traduzir a expressão, embora soe um pouco estranha em nosso idioma, especialmente no plural.

Figura 2: Árvore de busca com fator de derivação igual a 3

Em termos lógicos, portanto, a abordagem computacional dos problemas de busca é simples, pois se reduz a percorrer todos os caminhos possíveis entre o estado inicial e os possíveis estados terminais. Conhecidos os caminhos, basta escolher aquele que proporciona maior retorno, de acordo com o *critério de sucesso* estabelecido para o problema. Critério de sucesso, aqui, é o conjunto de regras que faz a diferenciação entre estados mais ou menos desejáveis. No jogo-da-velha, por exemplo, é desejável um estado terminal em que o agente consegue alinhar três marcas no tabuleiro de jogo. Ao contrário, é indesejável um estado terminal onde o adversário do agente consegue fazer o mesmo. Um estado terminal em que nenhum dos oponentes alinha três marcas corresponde a um meio-termo: é melhor do que perder, e pior do que vencer. Com base nesse critério, o agente buscará vencer o jogo, e na impossibilidade de fazê-lo, tentará um empate.

A escolha das jogadas, em sua forma básica, é feita por meio de um algoritmo chamado mini-max. O algoritmo tem esse nome porque ele busca maximizar os ganhos do agente, considerando, em contrapartida, que o adversário sempre agirá de modo a minimizar esses ganhos. Primeiro, são calculadas todas as variantes de jogo, até uma determinada profundidade da árvore de busca. Então, verifica-se o valor relativo de cada um dos estados a que se chega³⁸, nessa análise. Para cada nível acima, na árvore, procede-se da seguinte maneira: se em dado nível, cabe ao agente jogar, escolhe-se a jogada que conduz, no nível logo abaixo, à posição com avaliação mais favorável (max). Mas, se a jogada cabe ao adversário, parte-se do princípio que ele escolherá a jogada que minimiza os ganhos do agente. O algoritmo é aplicado recursivamente, das folhas à raiz da árvore. Com isso, se a avaliação das posições, nas folhas, for correta, o lance escolhido, na raiz, será o que oferece à máquina as melhores possibilidades de ganho, e ao adversário, as menores oportunidades de melhorar a sua situação.

A figura abaixo ilustra a escolha de uma jogada, com base no algoritmo mini-max. O jogador A (representado pelos quadrados brancos), tem três lances à sua disposição, **a**, **b** e **c**, e para cada um deles, o jogador B (representado pelos quadrados cinzas) dispõe de três respostas. No último nível da árvore, cada posição recebe um escore, de acordo com um critério de avaliação pré-definido. Um escore positivo significa uma posição favorável a A, e um escore negativo, uma posição favorável a B. O algoritmo, representado pelo jogador A, “supõe” que o jogador B tentará maximizar o seu próprio

³⁸ Esta avaliação corresponde a uma quantificação de fatores posicionais relevantes, feita pelas funções de avaliação heurística, conforme explicado na seção 3.5.

escore, e minimizar o escore de **A**. Então, na posição intermediária da árvore, o escore atribuído a **B** é o melhor, dentre suas possibilidades de escolha.

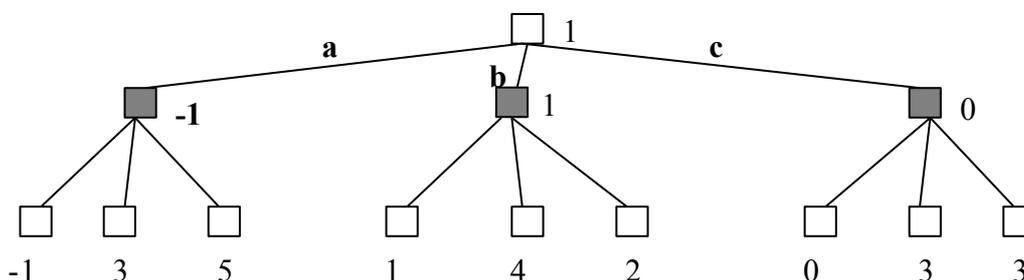


Figura 3: Escolha de um lance pelo algoritmo mini-max

Na situação representada, algoritmo escolherá o lance **b**, porque, consideradas as melhores respostas do jogador **B**, este é o lance que proporciona ao jogador **A** o maior escore.

Seguindo o esquema exposto, qualquer computador pessoal moderno pode realizar a análise de todas as variantes possíveis do jogo-da-velha, em uma fração de segundo, de forma que a aplicação do algoritmo mini-max proporciona um jogo perfeito. Basta atribuir, a cada posição final da análise um valor que traduz a situação alcançada, por exemplo, 10 para uma vitória do agente, zero para um empate e -10 para uma vitória do adversário.

Um aspecto importante do algoritmo mini-max é que ele não prevê blefes, ou seja, o agente nunca escolherá um lance na expectativa de que o adversário cometa um erro. Implicitamente, parte-se do princípio de que a capacidade de análise do agente é compartilhada por seu adversário. Pode-se dizer, portanto, que ao aplicar o algoritmo mini-max, para escolher os seus lances, o agente simula um jogo contra si mesmo³⁹.

A exploração de todos os caminhos de jogo, conforme já foi discutido, é possível para jogos desinteressantes, como o jogo-da-velha, mas impraticável para os demais jogos. Isso acontece porque o algoritmo de busca, na sua expressão mais elementar, tem complexidade exponencial. Isso quer dizer que o tempo necessário para explorar os caminhos de jogo possíveis é proporcional a K^n , onde K é o número médio de jogadas legais disponíveis para cada jogador, em cada posição de jogo, e n é o número de

³⁹ Diferentemente do analista humano, que tenta inferir as próximas jogadas com base no seu conhecimento sobre o adversário, e não com base no seu conhecimento sobre si mesmo.

meias-jogadas que se deseja prever. A meia-jogada é a jogada feita por um jogador, sem levar em conta a resposta adversária.

Vale notar que, no mundo dos jogos, o fator de derivação é sempre um valor médio aproximado, já que o número de jogadas legais disponíveis em cada posição varia muito no decorrer da partida. No xadrez, há posições com mais de cinquenta jogadas válidas, e outras com apenas uma ou duas. Em média cada posição do xadrez, no decorrer de uma partida, proporciona de 30 a 40 lances válidos⁴⁰. Conseqüentemente, a análise exaustiva de cinco lances à frente requer, de acordo com as estimativas mais otimistas, aproximadamente 30^{10} , ou 6×10^{14} posições⁴¹. Considerando o mais poderoso computador de xadrez já construído, o Deep Blue, analisando cerca de 200 milhões de posições por segundo, tal análise levaria cerca de 820 horas. Mas, se estendêssemos a análise exaustiva até o décimo lance à frente, em uma posição típica (totalizando 30^{20} posições), ela se arrastaria por aproximadamente 55 milhões de milhões de anos.

Além do tempo necessário para o cálculo das variantes, o espaço de armazenamento requerido seria também proibitivo, pois cada variante teria que ser armazenada na memória, até que todas fossem calculadas, para só então ser feita a escolha do lance mais adequado. Somente para se armazenar o resultado da busca com profundidade de cinco meias-jogadas, conforme estimativa acima, seriam necessárias, no mínimo, aproximadamente 10^{11} posições de memória, que corresponde a praticamente todo o espaço de armazenamento disponível na versão do Deep Blue que venceu Kasparov, em 1997, incluindo seus trinta discos rígidos de 4Gb, e seus trinta módulos de memória RAM, com 1Gb cada um.

No Go, a situação é pior do que no xadrez, já que o número de lances possíveis em cada posição, é muito maior: Ginsberg (1998, p.89) fala em 250 lances válidos por posição, mas Russel e Norvig (1995, p.137) elevam a estimativa a 360. Com isso, a análise exaustiva de variantes está fora de questão, para o Go, pois seriam 360^8 , ou seja, $1,5 \times 10^{19}$ caminhos a analisar, considerando apenas quatro lances, ou oito meias-jogadas à frente. Situação bem diversa ocorre no jogo de damas, onde os movimentos das peças

⁴⁰ As estimativas variam. Hsu e al. falam em 38 lances (1990, p.19), Ginsberg fala em 30 lances (1998, p.89) e Russell e Norvig falam em 35 (1995, p.129).

⁴¹ Como cada lance completo requer que cada um dos oponentes jogue, o expoente para cálculo das variantes dobra: 5 lances, expoente 10, 6 lances, expoente 12, e assim por diante. Generalizando, o número de posições a considerar em uma análise exaustiva, para jogos como xadrez, damas e Go, é da ordem de K^{2n} , onde K é o número médio de lances válidos, em cada posição, e n é o número de lances que se deseja “ver” à frente.

são muito limitados, e as regras freqüentemente obrigam o jogador a jogar um lance específico. Com isso, o número médio de lances válidos cai bastante, em relação ao Go e ao xadrez, de forma que a análise exaustiva de variantes, pelos modernos computadores, é mais bem sucedida no jogo de damas. Mesmo assim, a força bruta não basta para que os computadores joguem damas tão bem quanto os melhores jogadores humanos, particularmente no tabuleiro de cem casas (10x10)⁴².

Na prática, uma partida de xadrez ou Go não dura apenas cinco ou dez lances, mas trinta, quarenta, cinquenta lances ou mais. Portanto, é inútil qualquer tentativa de esgotar a análise desses jogos através do cálculo exaustivo de variantes. Em outras palavras, é impossível, na prática, a construção de uma árvore de busca que inclua todos os caminhos de jogo possíveis, do início ao fim de uma partida de xadrez ou Go. O mesmo pode ser dito para muitos outros jogos, especialmente os jogos probabilísticos e os de informação imperfeita, onde a tentativa de se prever todas as possibilidades de jogo freqüentemente levaria a uma função exponencial com elevadíssimo fator de crescimento.

Tomemos como exemplo o jogo de copas, que graças sistema operacional Windows[®] se tornou muito conhecido. Nesse jogo, o objetivo de cada agente é acumular o menor número possível de pontos, evitando, a cada “mão”, levar para si as cartas de copas e a dama de espadas. No início do jogo, os quatro jogadores têm, cada um, treze cartas na mão. O jogador **A** só sabe com certeza quais são as suas próprias cartas, num total de treze, mas poderia tentar otimizar sua estratégia, prevendo cada uma das possibilidades de arranjos de cartas nas mãos dos seus oponentes. Contudo, com as 39 cartas restantes distribuídas entre os adversários, são possíveis aproximadamente oito bilhões de combinações no início de cada rodada⁴³. Mesmo que, no decorrer do jogo, as cartas que cada adversário possui possam ser inferidas com razoável precisão, seria inviável otimizar uma estratégia para os lances iniciais, a partir de uma análise exaustiva de possibilidades. Isso significa que, para abordar algoritmicamente o jogo de copas, assim como o xadrez, o Go, e outros, são necessárias estratégias complementares à busca exaustiva, devido à impossibilidade prática de se esgotar a análise de todas as variantes possíveis.

⁴² Os campeonatos oficiais de damas são disputados em tabuleiros de 100 casas, ao contrário da versão mais popular do jogo, no Brasil, que utiliza o tabuleiro de 64 casas.

⁴³ O baralho tem 52 cartas. Dessas, 13 estão na mão do agente. Ao segundo jogador, também cabem 13 cartas, retiradas das 39 restantes, isto é, são ${}_{39}C_{13}$ combinações de “mãos” para o adversário **B**. Ao terceiro e quarto jogadores, cabem mais 13 cartas para cada um, retiradas das 26 que restam, e portanto, são mais ${}_{26}C_{13}$ combinações de “mãos” para os adversários **C** e **D**.

3.5. Buscas otimizadas

Nesse ponto, fica claro que os jogos realmente interessantes não podem ser abordados somente através dos algoritmos de busca, na sua versão elementar. Isso não significa, porém, que os algoritmos de busca sejam de todo ineficazes. Ao contrário, a otimização desses algoritmos tem sido responsável, em grande medida, pelo sucesso dos programas em vários jogos clássicos, e especialmente no xadrez, que nos interessa mais de perto. Isso ocorre porque o algoritmo mini-max não é usado pelos programas de jogos na sua forma pura, mas serve de base para a implementação de algoritmos mais elaborados. As ações dos programadores para tornar seus programas mais fortes seguem, basicamente, três vias complementares:

- (1) Otimizar o algoritmo de busca, propriamente dito, a fim de que o maior número possível de variantes do jogo possam ser testadas dentro dos limites de tempo e espaço de armazenamento disponíveis.
- (2) Tornar o algoritmo de busca mais seletivo, isto é, diminuir o fator de derivação da árvore de busca, de modo a aumentar o número de jogadas à frente que se pode pesquisar. Em outras palavras, diminuir a largura para aumentar a profundidade da árvore de busca.
- (3) Utilizar funções heurísticas para identificar as posições de jogo mais ou menos promissoras, a fim de proceder escolhas razoáveis, mesmo na impossibilidade de levar a cabo a análise de todas as variantes do jogo, até os estados terminais.

O item (1) traduz uma questão estritamente técnica, no domínio dos algoritmos e do hardware. Um mesmo programa, rodando em máquinas, distintas terá performances distintas, porque a performance está diretamente ligada ao número médio de lances que o programa “vê” adiante, que por sua vez, é diretamente dependente da velocidade com que o algoritmo de busca é executado. Por outro lado, dois algoritmos de busca, ainda que idênticos em termos lógicos, podem ser implementados de forma diferente, com reflexos marcantes no seu desempenho. Como afirmam Russel e Norvig (1995, p.123), “...um programa de xadrez que é 10% menos eficiente na utilização do tempo disponível será derrotado convincentemente, considerando-se os demais fatores nivelados.”⁴⁴

⁴⁴ “...a chess program that is 10% less effective in using its available time will be beaten into the ground, other things being equal.” A diferença de performance entre programas rodando em diferentes tipos de hardware tem sido constatada em inúmeros experimentos, muitos dos quais relatados informalmente no

Os itens (2) e (3) estão fortemente relacionados, porque, para selecionar algumas variantes de jogo em detrimento de outras, durante a busca, é necessário que o programa possa avaliar as posições atingidas ao cabo das variantes analisadas. O objetivo descrito em (2), ou seja, o estreitamento da árvore de busca, normalmente é alcançado através de uma técnica chamada *alpha-beta pruning*. Esta expressão (que será mantida no idioma inglês) indica um certo tipo de *poda* da árvore, com base em gatilhos de avaliação posicional, conforme os resultados da função de avaliação heurística.

Suponhamos que, ao analisar uma determinada posição, em que cabe ao agente jogar, existam apenas duas alternativas, A e B, e para cada uma delas, o adversário tenha cinco respostas possíveis, $A_{1..5}$, $B_{1..5}$. Esse cenário implicaria, em princípio, em dez variantes a analisar. Contudo, imaginemos que, ao se analisar a alternativa A, constata-se que qualquer resposta do adversário deixa o agente em situação favorável e, ao se analisar a alternativa B, constata-se que a primeira resposta considerada para o adversário, digamos, B_1 , deixa o agente em situação inferior. Ora, nesse caso é inútil prosseguir avaliando as alternativas B_2 até B_5 , porque, claramente, o lance A é o mais favorável para o agente, porque não dá chances ao adversário. Aplicando-se esse raciocínio a cada nível da árvore de busca, pode-se proceder uma enorme redução no número de variantes a analisar, o que significa, para os limites de tempo e de memória estabelecidos, a possibilidade de analisar a árvore em maior profundidade.

Uma árvore cujo fator de derivação é 36 produz mais de 60 milhões de variantes, quando analisada até uma profundidade de cinco meias-jogadas. Contudo, se o fator de derivação for reduzido para 6, o mesmo número de variantes é produzido pela análise da árvore até uma profundidade de dez meias-jogadas. Tal situação, que grosso modo caracteriza os ganhos alcançados pelos programas de xadrez, demonstra a importância do algoritmo de poda, para tornar os programas mais eficientes: sem a poda, uma série de jogadas irrelevantes são analisadas, consumindo um tempo precioso do hardware, que deveria ser usado para calcular, mais a fundo, as variantes que realmente interessam.

A redução do fator de derivação, portanto, é um aspecto crucial do aperfeiçoamento dos programas de jogos. Contudo, há aqui uma dificuldade a ser enfrentada, que é o estabelecimento de critérios confiáveis para a análise das posições alcançadas durante a busca. Naturalmente, se os critérios de análise forem ruins, a seleção das variantes mais

fórum de discussão “Computer Chess Club” (acesso disponível para membros em www.icdchess.com/cc.html).

promissoras será distorcida, com um impacto negativo sobre a qualidade dos lances escolhidos. É certo que, em alguns casos, há variantes claramente indesejáveis, porque conduzem à perda do jogo. Mas, na maioria das vezes, a distinção entre posições mais ou menos desejáveis não é tão evidente.

Por um lado, a escolha dos melhores lances requer o máximo aprofundamento possível da árvore de busca. Por outro lado, o aprofundamento da árvore de busca pressupõe algum critério para escolher determinadas variantes, em detrimento de outras. Esse aparente impasse é resolvido através de uma técnica conhecida como busca heurística, que, com base em critérios gerais de análise (heurísticas), associados à busca propriamente dita, tenta encontrar *soluções ótimas* onde a busca por *soluções perfeitas* é inviável, devido a limitações práticas no processamento de algoritmos.

Na sua definição de heurística, Rich e Knight exprimem este ponto claramente:

Uma heurística é uma técnica que melhora a eficiência de um processo de busca, possivelmente sacrificando pretensões de completude. Heurísticas são como guias turísticos. Eles são bons na medida em que apontam para direções geralmente interessantes; são ruins na medida em que deixam de assinalar pontos de interesse para certas pessoas. Algumas heurísticas ajudam a guiar um processo de busca sem sacrificar quaisquer pretensões de completude que o processo pudesse ter. Outras (de fato, algumas das melhores) podem ocasionalmente fazer com que um caminho excelente passe despercebido. Mas, na média, elas melhoram a qualidade dos caminhos que são explorados. Usando boas heurísticas nós podemos esperar obter soluções boas (embora possivelmente não-ideais) para problemas difíceis, tais como o do caixeiro-viajante, em tempo menor do que o exponencial. (Rich and Knight, 1991, p.41)⁴⁵

⁴⁵ “A heuristic is a technique that improves the efficiency of a search process, possibly by sacrificing claims of completeness. Heuristic are like tour guides. They are good to the extent that they point in generally interesting directions; they are bad to the extent that they miss points of interest to particular individuals. Some heuristics help to drive a search process without sacrificing any claims to completeness that the process could have had. Others (in fact, many of the best ones) may occasionally cause an excellent path to be overlooked. But, on average, they improve the quality of the paths that are explored. Using good heuristics, we can hope to get good (though possibly nonoptimal) solutions to hard problems, such as the traveling salesman, in less than exponential time. (Rich and Knight, 1991, p.41)”

Nos problemas de busca, como os cálculos de variantes no xadrez, as heurísticas são aplicadas com a finalidade de atenuar os efeitos da complexidade computacional, direcionando a busca por caminhos que *provavelmente* são os mais promissores. O uso de heurísticas, em problemas de tal natureza, implica em um compromisso: abdica-se da exatidão que uma busca exaustiva poderia proporcionar, em prol da velocidade da busca, necessária para fazer face às limitações de tempo e espaço de armazenamento para o processamento dos algoritmos. Por isso, é necessário que as heurísticas sejam implementadas de forma eficiente, a fim de que se obtenha, através delas, o efeito esperado.

Suponhamos que um programador de xadrez dispõe de duas heurísticas, H_1 e H_2 , para avaliação posicional. Na decisão sobre qual das duas usar, ele levará em consideração dois fatores: por um lado, o nível de refinamento das heurísticas, que se reflete na sua eficácia em avaliar corretamente as posições e, por outro lado, a velocidade com que cada uma das heurísticas é processada. Se H_1 for cem vezes mais lenta do que H_2 , embora muito mais refinada, talvez H_2 seja a preferida do programador, porque a implementação de H_1 fará com que o algoritmo de busca tenha sua capacidade significativamente reduzida⁴⁶.

Como a velocidade no cálculo de variantes é crucial para o desempenho dos programas, talvez não compense reduzi-la, ainda que seja para usar uma heurística de qualidade superior. Em síntese, uma heurística aprimorada tende a tornar o cálculo de variantes mais eficaz, eliminando rapidamente, da árvore de busca, linhas de jogo não promissoras. Em contrapartida, uma heurística melhor geralmente requer mais tempo para ser processada, e assim o tempo disponível para o cálculo de variantes é reduzido. A opção entre heurísticas mais sofisticadas e lentas, ou heurísticas mais simples e rápidas, é um problema crítico do xadrez computacional, para o qual não se tem uma solução definitiva. Conseqüentemente, a fim de otimizar a performance dos seus programas, os programadores recorrem a testes numerosos e sucessivos refinamentos dos algoritmos, em busca do melhor balanço entre a velocidade de busca e a sofisticação heurística dos programas. Esses dois fatores, eficácia heurística e eficiência na busca, estão sempre em oposição, competindo pelo tempo de CPU disponível para

⁴⁶ Tendo em vista que o xadrez é sempre jogado contra o relógio, qualquer intervalo de tempo despendido com a avaliação posicional é subtraído ao cálculo de variantes. Estimando grosseiramente: se o fator de derivação médio da árvore de busca for igual a seis, ao multiplicar por cem o tempo gasto com a avaliação posicional estaremos reduzindo a profundidade média da árvore de busca em duas a três meias-jogadas, aproximadamente ($6^{2.57} \approx 100$).

processamento. Por isso a sintonia fina de um programa de xadrez é um trabalho minucioso, que se processa ao longo de anos.

O incremento de performance dos algoritmos de busca é uma questão eminentemente técnica, mas o aprimoramento das heurísticas é uma questão com um importante viés epistemológico. Esse ponto ficará claro mais adiante, na medida em que nos aprofundarmos na análise dos programas que jogam xadrez.

3.6. O xadrez como campo de provas

Os jogos têm sido um campo de grande interesse para a ciência da computação, desde os seus primórdios. Por um lado, o desafio de se construir programas capazes de enfrentar os seres humanos diretamente, no campo intelectual, é por si só cativante. Por outro lado, a construção de tais programas requer a aplicação de técnicas computacionais sofisticadas, que podem ser utilizadas com outros fins, na resolução de “problemas mais relevantes”, para usar a expressão de Shannon (1950).

Hubert Dreyfus (1992, pp.292-293) classifica o xadrez como uma atividade inteligente “complexa-formal” (complex-formal). Isso significa que o xadrez se esgota na sua lógica interna, definida por um conjunto de regras sem ambigüidade. Assim como os outros problemas nessa categoria, o xadrez poderia, em princípio, ser abordado através do cálculo exaustivo de possibilidades, o que levaria a um jogo perfeito. Porém, como já vimos, no decorrer deste texto (seção 3.2), a tentativa de esgotar a análise do xadrez, por via computacional, está fadada ao insucesso, tendo em vista o presente quadro de referências da física, que limita as possibilidades práticas da computação.

Eis como o próprio Dreyfus define a classe de problemas complexo-formais:

Ela envolve comportamentos que são em princípio formalizáveis, mas na verdade, intratáveis. Na medida em que o número de elementos cresce, o número de transformações cresce exponencialmente com o número de elementos envolvidos. Tal como usado aqui, o termo “complexo-formal” inclui aqueles sistemas que na prática não podem ser abordados através de algoritmos de enumeração exaustiva, e portanto, requerem programas heurísticos. (1992, p.223)⁴⁷

⁴⁷ It contains behavior that is in principle formalizable but in fact intractable. As the number of elements increases, the number of transformations required grows exponentially with the number of elements

Em última análise, a impossibilidade de se resolver o xadrez e problemas similares pelo cálculo exaustivo se deve a duas variáveis: tempo e meios de processamento. Para se esgotar computacionalmente o jogo de xadrez, em um tempo razoável, seriam necessários recursos computacionais impensáveis hoje em dia⁴⁸. Por outro lado, tendo em mente os recursos computacionais de que podemos esperar dispor, o tempo necessário para esgotar todas as possibilidades do jogo de xadrez seria várias ordens de grandeza superior à idade estimada do universo. Portanto, a afirmativa de que o xadrez, assim como problemas de mesma natureza, se esgotam na sua lógica interna, é verdadeira apenas enquanto tais problemas são considerados em seu caráter matemático abstrato. Em termos práticos, qualquer tentativa de resolvê-los esbarra nas contingências do mundo físico, que se manifesta no tempo e no poder de processamento disponíveis⁴⁹.

No universo dos jogos clássicos, o xadrez ocupa uma posição peculiar, sob uma perspectiva de ciência da computação. Ginsberg (1998) faz notar que, em alguns jogos, como as damas, o dominó e as palavras cruzadas (Scrabble⁵⁰), os programas baseados em técnicas de busca heurística, associadas a grandes bases de dados, superaram a capacidade humana, aparentemente de forma definitiva. Em outros jogos, como o Go e o pôquer, os melhores programas ainda têm desempenho muito aquém dos jogadores humanos de primeira linha. No caso do xadrez, a superação definitiva dos humanos

involved. As used here, 'complex-formal' includes those systems which in practice cannot be dealt with by exhaustive enumeration algorithms (chess, go, etc.), and thus require heuristic programs (1992, p.293).

⁴⁸ Não apenas *indisponíveis* hoje em dia, mas *impensáveis*, dentro das perspectivas científicas e tecnológicas atuais.

⁴⁹ E esse poder de processamento é determinado, em princípio, pela *quantidade de matéria* disponível para a construção dos computadores, que é finita.

⁵⁰ *Scrabble* é o nome patenteado para a modalidade de palavras cruzadas onde os adversários colocam sobre o tabuleiro, consecutivamente, letras de madeira ou plástico, na tentativa de acumular o maior número de pontos. Cada letra tem um número de pontos associado a ela, variando conforme a frequência relativa da letra nas palavras do idioma em que o jogo é comercializado: as letras mais frequentes pontuam menos, e as menos frequentes, pontuam mais. Cada jogador começa com sete letras, escondidas dos demais jogadores, e a cada vez que acrescenta letras nas palavras já formadas sobre o tabuleiro, o jogador marca os pontos correspondentes à nova palavra, e repõe as letras que usou, a partir de um monte de letras viradas de cabeça para baixo. Vence o jogador que tiver o maior número de pontos, quando as letras disponíveis não permitirem formar novas palavras, ou aquele que alcançar primeiro um certo número de pontos, combinado previamente.

pelas máquinas ainda não se deu, mas os melhores programas da atualidade já competem com sucesso contra jogadores de elite.

Nos jogos em que as máquinas têm vantagem decisiva, ela decorre da aplicação de heurísticas associada a algoritmos de busca poderosos, que se tornam tanto mais eficazes quanto menor é o número médio de jogadas permitidas em cada posição do jogo. Nos jogos em que o ser humano tem uma grande vantagem sobre as máquinas, a eficácia das técnicas de busca é reduzida, ou porque o número médio de jogadas possíveis em cada posição é muito grande, ou porque parte significativa dos dados não está disponível para análise, no processo de escolha das jogadas (jogos de informação imperfeita). O xadrez computacional é uma espécie de meio-termo: as técnicas de busca são comprovadamente eficazes para aumentar o padrão de jogo dos programas, mas por si mesmas são insuficientes para que os programas atinjam o nível dos grandes mestres humanos. Por outro lado, o esforço concentrado de programadores e jogadores de xadrez, ao longo de quase cinco décadas, produziu heurísticas muito refinadas, que compensam, em grande medida, as limitações práticas dos algoritmos de busca.

Nenhum outro jogo tem tantos adeptos quanto o xadrez, nem tampouco a mesma quantidade de bibliografia específica. Isso faz do xadrez é um *campo de provas* fecundo para as técnicas computacionais, porque elas podem ser testadas no confronto com um considerável corpo teórico, aperfeiçoado no decorrer de séculos. Assim, o desempenho de um programa de xadrez não é medido apenas por seus resultados diante de jogadores humanos, sempre sujeitos a variáveis psicológicas, que em tese, poderiam comprometer a objetividade da avaliação. Para medir o desempenho de um programa, procede-se também à comparação entre as soluções encontradas por ele e as soluções encontradas por enxadristas⁵¹ consagrados, para problemas específicos.

A competição entre computadores e jogadores humanos, ou entre diferentes computadores, permite a avaliação empírica dos méritos e deficiências dos algoritmos usados nos programas. No xadrez computacional, a busca pela excelência faz com que os programadores lancem mão de toda a sua técnica e criatividade, explorando ao máximo as possibilidades do hardware disponível. No decorrer de muitas partidas, jogadas sob condições controladas, a análise do comportamento dos programas proporciona elementos para o seu aperfeiçoamento.

⁵¹ No presente texto, a palavra enxadrista, assim como a palavra jogador, sempre se referem a seres humanos. Eventualmente, utiliza-se a expressão “jogador humano”, para assegurar a clareza de uma ou outra passagem.

Questões muito genéricas, sobre a possibilidade das máquinas imitarem a inteligência humana, têm sido formuladas, sem respostas satisfatórias. Ao lidar com essas questões, contudo, vale lembrar o alerta de Weizenbaum: “Inteligência é um conceito sem sentido, em si mesmo. Ele requer um quadro de referência, a especificação de um domínio de pensamento e ação, a fim de dar-lhe significado (1976, p.204-205).” Portanto, ao se discutir as possibilidades relativas de humanos e computadores, seria interessante fazê-lo em domínios bem definidos.

O xadrez é um domínio estritamente formal, e o desenvolvimento de uma partida, em tese, independe de qualquer fator externo a ela própria, embora, na prática, fator tempo seja determinante para a escolha das jogadas. Mas, exceto por esse detalhe, as máquinas que jogam xadrez só precisam se relacionar com o ambiente através de uma interface muito simples, e também podem permanecer alheias a quaisquer variáveis de cunho cultural, lingüístico, etc. Assim, a obtenção de um desempenho notável no jogo de xadrez, pelas máquinas, não envolve a simulação dos aspectos da inteligência humana ligados ao contexto ambiental. Com isso, espera-se que a supremacia da máquina sobre o ser humano, no xadrez, seja mais fácil de alcançar do que em domínios abertos, como a interpretação de textos, compreensão da fala ou a livre movimentação no espaço físico.

No âmbito humano, o jogo de xadrez está inserido em um vasto universo de discurso, onde as habilidades dos jogadores se desenvolvem e se exprimem. A performance dos grandes jogadores, ao resolver problemas específicos, nas suas análises e partidas, se transforma em matéria de discurso, que uma vez sistematizado, contribuirá para aumentar as habilidades de outros jogadores, através de livros palestras, e outros recursos pedagógicos. Uma parte do *discurso enxadrístico* é traduzida em regras e fórmulas matemáticas, passando a compor as heurísticas de avaliação posicional usadas nos programas. Uma vez que as heurísticas são de fundamental importância para que os programas joguem bem, um dos aspectos mais relevantes na análise do xadrez computacional é a possibilidade de se exprimir por via heurística o discurso sobre o xadrez – o que implica em se discutir, também, a possibilidade de se traduzir toda a habilidade dos maiores jogadores por meio de um discurso sistemático.

Os pesquisadores do xadrez computacional são unânimes em considerar que as heurísticas atuais ainda estão muito longe de simular o conhecimento enxadrístico de um enxadrista de elite. Se esses mesmos jogadores encontram dificuldades diante dos programas, é porque a abordagem computacional do jogo é radicalmente diferente da abordagem humana, e não porque os programas *entendem* mais de xadrez do que os

grandes mestres. Os programas *jogam* muito bem o xadrez, principalmente porque calculam com grande exatidão um número enorme de variantes, muito além das possibilidades dos humanos. Ainda assim, a avaliação posicional dos grandes jogadores é tão elaborada, que eles conseguem, a seu turno, fazer face ao poder de cálculo das máquinas. É essa tensão que torna o xadrez computacional tão atraente: aos programadores, permite-se a utilização de todos os recursos para superar a performance humana em um domínio totalmente formal, amplamente estudado, e onde a associação de grandes bases de dados, com o poder de cálculo das máquinas tem um efeito decisivo no desempenho dos programas.

Para os jogadores humanos, o xadrez envolve fatores como fadiga e lapsos de memória, aos quais a máquina é totalmente alheia – e ainda assim, os humanos continuam resistindo ao poder de cálculo dos computadores. Ao discutir as razões desse impasse, no duelo entre homens e máquinas, esta dissertação enfatiza os fatores que limitam o desempenho da máquina, no âmbito epistemológico. O caminho complementar, de ênfase nas possibilidades dos enxadristas humanos, é uma tarefa altamente relevante que, contudo, foge ao escopo desta dissertação.

É possível que o avanço do hardware e das técnicas de programação aumente tanto a eficiência dos algoritmos de busca que, em alguns anos, os programas de xadrez superem em definitivo os enxadristas humanos, em jogos de torneio, através do desenvolvimento da força bruta a níveis extremos. Mas, nem tudo no xadrez se resume ao sucesso em jogos de competição. Além disso, como a força bruta é insuficiente para resolver a maior parte dos problemas relevantes com que a IA se defronta, persistirá a necessidade de se investigar até que ponto a inteligência pode ser simulada por meios algorítmicos. Nesse contexto, as heurísticas têm um papel central, e o xadrez continuará sendo um campo privilegiado de análise.

4. Xadrez Computacional

4.1. Um pouco de história

Ainda na primeira metade do século XIX, Charles Babbage mencionou, no texto “The Life of a Philosopher”, a possibilidade de programar a sua Máquina Analítica para jogar xadrez (Levy e Newborn, 1991, p.25-26). Mas, como essa Máquina jamais saiu do papel, a idéia, que naturalmente não poderia ser implementada com a tecnologia da época, não chegou a ser desenvolvida em maiores detalhes. Vale, contudo, acrescentar, que Babbage aparentemente subestimou a quantidade de operações que seriam necessárias para a execução do programa proposto.

Em 1914, o Engenheiro espanhol Torres y Quevedo apresentou a primeira máquina capaz de tratar automaticamente certas posições do jogo de xadrez. Especificamente, o Autômato de Quevedo era capaz de dar um xeque-mate elementar, de rei e torre contra rei sozinho, sem a assistência humana, partindo de qualquer posição legal envolvendo estas três peças. Provavelmente, o Autômato de Quevedo foi a primeira máquina a incorporar o conceito de desvio condicional, idealizado por Babbage em meados do século XIX. Uma vez que o cada lance feito pela máquina, até o xeque mate, variava em função do lance anteriormente realizado pelo adversário, o Autômato tinha que aplicar desvios condicionais para determinar suas jogadas: não havia um caminho fixo a seguir, mas vários caminhos possíveis, em função das jogadas adversárias.

Apesar de bastante avançado para a época, o Autômato de Quevedo incorporava artifícios técnicos específicos para resolver um problema restrito, e não um conjunto de procedimentos gerais para jogar xadrez. Tendo em vista suas limitações de ordem prática, mas, sobretudo, de ordem conceitual, não pode ser encarado como uma máquina de jogar xadrez, à semelhança das atuais. Os modernos computadores, programados para jogar xadrez, baseiam-se em algoritmos de busca e de análise posicional, que não se encontram presentes na concepção do Autômato de Quevedo.

Foi o criador da moderna teoria da informação, Claude Shannon (1950), quem primeiro descreveu, em detalhes, um algoritmo para jogar xadrez, voltado para os computadores de uso geral (“general purpose computers”, nas palavras de Shannon). Alan Turing também se interessou pela questão, mais ou menos na mesma época que Shannon, e desenvolveu idéias semelhantes, mas o artigo de Shannon é o ponto de partida para o desenvolvimento teórico do xadrez computacional, porque discute pela primeira vez,

sistematicamente, questões que viriam a nortear muitas das pesquisas subseqüentes no ramo.

Como engenheiro, Shannon interessava-se pela aplicação da teoria à solução de problemas práticos. Fiel a essa postura, ele inicia o seu artigo dizendo⁵²:

Este artigo trata do problema de se construir uma rotina de computador, ou ‘programa’ para um moderno computador de uso geral, que o tornará capaz de jogar xadrez. Embora talvez não tenha importância prática, a questão tem um interesse teórico, e *espera-se que uma solução satisfatória para este problema atue como um suporte para atacar outros problemas de natureza similar, e mais significativos.* (grifo meu)

Alguns dos problemas “de maior relevância”, a que Shannon se reporta no artigo, são o cálculo simbólico, o projeto de circuitos eletrônicos e a tradução automática entre idiomas. O xadrez seria, então, um excelente banco de ensaios para métodos computacionais sofisticados, destinados a resolver automaticamente problemas que envolvem raciocínio. O artigo de Shannon foi escrito no período em que apareceram os primeiros computadores eletrônicos digitais, e mais de seis anos antes da expressão “inteligência artificial” ter sido criada. Por isso, as seguintes palavras têm um significado especial:

A máquina de xadrez é ideal para se começar, uma vez que: (1) o problema está claramente definido, tanto em termos das operações permitidas (as jogadas) como em termos do objetivo (o xeque-mate); (2) ele não é nem tão simples a ponto de ser trivial nem difícil a ponto de não se encontrar uma solução satisfatória; (3) geralmente considera-se que jogar bem o xadrez requer ‘pensamento’; uma solução para este problema nos forçará ou a admitir a possibilidade do pensamento mecanizado, ou a restringir ainda mais nosso conceito de ‘pensamento’; (4) a estrutura discreta do xadrez casa bem com a estrutura digital dos computadores modernos (shannon, 1950).⁵³

⁵² This paper is concerned with the problem of constructing a computing routine or ‘program’ for a modern general-purpose computer which will enable it to play chess. Although perhaps of no practical importance, the question is of theoretical interest and it is hoped that a satisfactory solution of this problem will act as a wedge in attacking other problems of similar nature and of greater significance (grifo meu).

⁵³ The chess machine is an ideal one to start with, since: (1) the problem is sharply defined both in allowed operations (the moves) and in the ultimate goal (checkmate); (2) it is neither so simple as to be

Assim, o xadrez foi reconhecido como um problema relevante para a inteligência artificial, antes mesmo que ela existisse como disciplina independente. De fato, o interesse pelo xadrez, nos círculos da computação, nunca decresceu, e muitos recursos foram investidos na tentativa de se criar máquinas capazes de bater grandes jogadores humanos – até a década de 1970, sem qualquer perspectiva de retorno comercial direto.

A partir dos anos sessenta, as máquinas de xadrez começaram a ser postas à prova contra adversários humanos, inicialmente com resultados modestíssimos, pois elas jogavam como principiantes (Levy e Newborn, 1991). A partir dos anos setenta, torneios entre computadores passaram a ser promovidos assiduamente. Aos poucos, a evolução dos algoritmos e do hardware foi fazendo com que os programas galgassem degraus cada vez mais expressivos. Em 1978, pela primeira vez, um programa de computador, o CHESS 4.7, empatou com um mestre internacional (MI) de xadrez em uma partida de torneio, e depois o derrotou, em uma outra partida no mesmo match. O match, ao final, foi vencido pelo MI escocês David Levy, ele próprio um entusiasta do xadrez computacional (Levy e Newborn, 1991, pp.92-102).

Em meados dos anos oitenta, com o advento dos circuitos eletrônicos de altíssima escala de integração (VLSI, ou Very Large Scale Integration), surgiu uma nova geração de máquinas de xadrez, baseadas em hardware específico para a geração de lances e cálculo de variantes. Essa tendência, paralelamente ao uso de supercomputadores para rodar programas de xadrez, marca a predominância do poder computacional como forma de aumentar o nível de jogo das máquinas. Em 1986, contudo, o programa REBEL, rodando em um modesto microprocessador de 8 bits, defrontou-se satisfatoriamente com programas que rodavam em máquinas milionárias (Levy e Newborn, 1991, p.122), e com isso demonstrou que heurísticas sofisticadas podem, até certo ponto, se contrapor à força bruta. Em 1989, o Deep Thought confirmou o novo patamar do xadrez computacional, ao vencer um forte torneio na Califórnia, após derrotar dois grandes mestres consagrados, Tony Miles e Bent Larsen, este último um ex-candidato ao título mundial. Durante os anos noventa, além dos celebrados confrontos entre Kasparov e o Deep Blue, o xadrez computacional se beneficiou do contínuo progresso dos programas para microcomputadores, cuja velocidade de

trivial nor too difficult for satisfactory solution; (3) chess is generally considered to require “thinking” for skillful play; a solution of this problem will force us either to admit the possibility of mechanized thinking or to further restrict our concept of “thinking”; (4) the discrete structure of chess fits well in the digital structure of modern computers (Shannon, 1950).

processamento e capacidade de armazenamento tornou-se comparável às dos supercomputadores do início dos anos oitenta.

As competições entre seres humanos e computadores prosseguem despertando muito interesse da comunidade enxadrística. Ao mesmo tempo, os computadores se transformaram em instrumentos indispensáveis de treinamento para todos aqueles que levam o jogo a sério. Além disso, já começam a ser ensaiadas competições em que os humanos jogam auxiliados por computador, à maneira das equipes de automobilismo, em que o talento do homem e a força da máquina são ambos essenciais para a vitória. No xadrez postal⁵⁴, esta já é uma prática corriqueira, há vários anos.

4.2. O estado da arte

Os melhores programas de xadrez baseiam-se em uma abordagem pragmática, que consiste na utilização de todas as técnicas disponíveis para maximizar sua performance, sem a preocupação de aproximá-los da forma de pensar dos seres humanos. A velocidade de processamento das máquinas permite o cálculo preciso de inúmeras variantes de jogo, enquanto sua capacidade de armazenamento de dados permite o uso de grandes *bibliotecas*⁵⁵ de aberturas e finais, para serem consultadas durante os jogos.

Em linhas gerais, os programas de xadrez têm três módulos:

- (1) Um módulo gerador de jogadas, que especifica todos os lances legais em uma posição. Esta operação é automática e inconsciente para os jogadores humanos, e não requer deles nenhum tempo de análise. Em termos computacionais, contudo, a geração de lances legais, de forma eficiente, é por si só um problema relevante.
- (2) Um módulo calculador de variantes, que especifica os possíveis caminhos que o jogo pode tomar, de acordo com as jogadas do programa e as respostas do adversário.

⁵⁴ Modalidade muito antiga, em que os lances são trocados por correspondência, de modo que uma partida pode durar vários meses, ou mesmo anos. Atualmente, é comum a troca de lances através de correio eletrônico.

⁵⁵ Na terminologia do xadrez computacional, uma biblioteca é um arquivo que contém posições típicas de aberturas (ou finais), juntamente com as jogadas recomendadas em cada uma delas, de acordo com a experiência adquirida em jogos e análises anteriores.

- (3) Um módulo de avaliação de posições, que aplica heurísticas para fazer a distinção entre posições mais ou menos promissoras, reduzindo o número de variantes a serem testadas.

Esses módulos básicos compõem aquilo que se convencionou chamar de “engine”, o *motor* ou núcleo de um programa de xadrez. Contudo, os bons programas de xadrez sempre integram outros recursos para aumentar a qualidade do seu jogo, em especial as bibliotecas de aberturas, ou seja, conjuntos de posições típicas, do início da partida até o décimo, vigésimo, ou mesmo trigésimo lance. Os lances armazenados na biblioteca são selecionados cuidadosamente, a partir de partidas disputadas por jogadores fortes, ou a partir de análises teóricas demoradas. Para fins de treinamento, os usuários dos programas podem escolher que aberturas ensaiar, mas, para efeito de competição, os programas são direcionados para buscar as posições de abertura mais adaptadas ao seu modo de jogar. Na grande maioria das vezes, portanto, um bom programa não analisa as posições iniciais do jogo, mas recupera os lances diretamente, a partir de uma biblioteca de aberturas que contém milhões de posições previamente testadas. As bibliotecas de abertura são muito importantes, pois os módulos de análise dos programas freqüentemente encontram dificuldades em efetuar os primeiros lances do jogo de forma consistente. Além disso, o uso de bibliotecas garante a possibilidade de se variar os lances, mas com a segurança da experiência adquirida em jogos anteriores. Contudo, a elaboração das bibliotecas é também um processo propenso a erros, e algumas derrotas de programas fortes têm sido atribuídas ao caráter desfavorável de posições escolhidas a partir das suas bibliotecas.

Com o advento do CD-ROM e das memórias RAM⁵⁶ de alta capacidade, as bibliotecas de finais também se tornaram comuns. Elas funcionam de modo análogo às bibliotecas de aberturas, mas entram em ação quando há um número de peças muito reduzido no tabuleiro. Tradicionalmente, os computadores eram deficientes no jogo com poucas peças, porque, nessas condições, tanto a força bruta quanto às heurísticas são freqüentemente insatisfatórias. Uma solução encontrada para esse tipo de problema foi à análise reversa, ou seja, a busca de lances de trás para frente, iniciando de uma posição final, de xeque-mate ou empate por “afogamento”⁵⁷. Essa técnica, que consome muito

⁵⁶ Memórias de acesso aleatório e resposta rápida. Atualmente, os dados armazenados em RAM podem ser recuperados em intervalos de tempo da ordem de nanossegundos. Em comparação, os dados armazenados em disco magnético ou CD-ROM só podem ser recuperados em intervalos de milissegundos.

⁵⁷ O empate por “afogamento” ocorre, no xadrez, quando, cabendo a um dos adversários jogar, ele não dispõe de nenhum lance legal, mas o seu rei não se encontra em xeque. É uma situação diferente da que

tempo e memória, não poderia ser aplicada em tempo real, durante os jogos. Contudo, ela permitiu a criação de bases de dados onde são armazenadas todas as combinações relevantes de até cinco peças sobre o tabuleiro, juntamente com os lances a serem realizados em cada situação. Dessa forma, os computadores jogam os finais previamente analisados de forma matematicamente perfeita: eles explorarão todas as chances de vitória, nunca perderão uma posição em que há empate certo, e empatarão em posições teoricamente perdidas, sempre que lances inexatos dos adversários o permitirem⁵⁸.

Uma outra técnica para melhorar o jogo das máquinas é denominada *extended book* (livro estendido), e foi usada no Deep Blue (Campbell et al., 2002). Nas palavras dos autores,

A idéia básica é sumarizar a informação disponível para cada posição de uma base de dados de 700.000 partidas, e usar a informação sumária para atrair o Deep Blue ao consenso em torno da teoria de aberturas (p.77).⁵⁹

O sumário das informações faz referência, por exemplo, aos resultados obtidos com cada jogada (supostamente, se a partida foi vencida, a jogada tem maiores chances de ser boa), a atualidade das jogadas (supostamente, quanto mais recente a partida referenciada, melhor, porque os lances se baseiam nos avanços da teoria), o número de vezes que cada jogada foi usada e o nível (rating) dos jogadores responsáveis pelas jogadas (supostamente, quanto maior o rating, melhor, porque jogadores de rating mais elevado tendem a fazer lances melhores). Com base em uma ponderação desses fatores, os lances disponíveis na biblioteca, para uma determinada posição, são comparados com os lances analisados pela máquina, escolhendo-se um lance da biblioteca, supostamente bom, a não ser que a rotina de avaliação encontre outro, muito melhor. Em suma, o programa se baseia nas escolhas dos grandes mestres humanos, a não ser que a análise

ocorre no jogo de damas, onde um jogador que não pode fazer nenhum lance é declarado perdedor.

⁵⁸ É interessante notar que, em termos puramente lógicos, a análise reversa poderia ser estendida indefinidamente, de forma a se catalogar todas as posições possíveis do jogo de xadrez, juntamente com as respectivas *jogadas ideais*. Teríamos, assim, para o xadrez, uma solução parecida com aquela proposta para o jogo-da-velha, na seção 3.2 deste texto. Contudo, dado o enorme número de posições do jogo de xadrez, não há meio de calculá-las, nem de catalogá-las todas, considerando-se os referenciais tecnológicos da atualidade.

⁵⁹ The basic idea is to summarize the information available at each position of a 700,000 game database, and use the summary information to nudge Deep Blue in the consensus direction of chess opening theory (p.77).

em tempo real, feita pelo próprio programa, revele algum lance aparentemente muito bom, ausente dos jogos da biblioteca. De acordo com Campbell, o extended book do Deep Blue foi importante no segundo jogo do match de 1997, em que a máquina derrotou Kasparov.

A força competitiva dos programas de xadrez, atualmente, é função da sua capacidade de calcular variantes, aliada à qualidade das heurísticas utilizadas para a avaliação posicional. A importância atribuída ao poder de cálculo dos programas é tanta que um dos principais indícios para se estimar a força dos programas são os “nós por segundo” (Nps, ou nodes per second), que é a medida de quantas posições distintas, em média, um programa, pode analisar, a cada segundo. Um mesmo programa, rodando em máquinas com diferentes capacidades de cálculo, torna-se sensivelmente mais forte ou mais fraco, devido à variação do fator Nps. Na literatura sobre xadrez computacional (Levy e Newborn, 1990; Hsu e al., 1990), a força bruta é tida como o principal fator para a melhoria da performance dos programas, e é significativo que o Deep Blue, única máquina a derrotar um campeão mundial, em condições de torneio, tenha sido também a mais rápida máquina de xadrez já construída.

Hsu e al. (1990) mostraram, com base na interpolação de dados históricos, que o rating performance dos computadores de xadrez vinha crescendo (até a época em que o referido artigo foi escrito) em relação direta com a capacidade das máquinas em antecipar variantes, medida em meias-jogadas. O crescimento apontado era então da ordem de 200 pontos de rating, para cada meia-jogada a mais que a máquina for capaz de analisar. Por exemplo, se uma máquina capaz de analisar quatro lances adiante (oito meias-jogadas) tivesse um rating de 2200, uma máquina capaz de analisar cinco lances a frente (dez meias-jogadas) deveria ter um rating-performance em torno de 2600. Enquanto a primeira teria desempenho comparável ao de amadores fortes, a segunda teria desempenho superior ao da maioria dos mestres internacionais, e mesmo ao de alguns grandes mestres.

Conforme já foi explicado no capítulo anterior, para analisar uma meia-jogada a mais, a máquina tem que ter o seu poder de processamento multiplicado por um fator K, que hoje em dia é estimado em aproximadamente 6 (Russel e Norvig, 1995, p.131). Por esta via, pode-se estimar que o rating dos programas cresce linearmente, com o crescimento exponencial do seu poder de cálculo. Para os programas de microcomputador, as

estimativas variam entre 40 a 70 pontos a mais de rating performance, cada vez que a sua velocidade de cálculo é dobrada⁶⁰.

Junto com a força bruta, a qualidade das heurísticas é o grande fator que diferencia os programas de xadrez uns dos outros. Todos os melhores programas comerciais embutem uma quantidade considerável de conhecimento⁶¹ enxadrístico, traduzido por meio de fórmulas e regras, usadas para a avaliação das posições que ocorrem nas partidas. Um exemplo marcante do poder das heurísticas superando a força bruta é a partida entre o microcomputador Mephisto e o Deep Thought, jogada em 1989, no 20th ACM Championship (Levy e Newborn, 1991, p.126-127). À época, o Deep Blue era capaz de analisar cerca de 700.000 Nps⁶². Em contrapartida, o Mephisto, baseado em um microprocessador de 8 bits, não poderia analisar mais do que 5.000 ou 10.000 Nps, ou seja, uma diferença da ordem de 100 vezes, a favor do poder de cálculo do Deep Thought. Ao vencer a partida o Mephisto demonstrou que a força bruta, sozinha, não determina a superioridade absoluta de um programa de xadrez sobre os demais.

Certamente, se o antecessor do Deep Blue tivesse jogado um match contra o Mephisto, teria vencido por uma larga margem. Contudo, o simples fato de um programa poder derrotar um outro, cem vezes mais rápido, demonstra que a força bruta não é tudo, e que as heurísticas são cruciais, especialmente em posições em que não há longas variantes forçadas, como aconteceu jogo em questão⁶³. Alguns dos programas para microcomputadores são, hoje em dia, provavelmente quase tão fortes quanto o Deep Blue II, embora com uma capacidade de cálculo cerca de cem vezes inferior. Isso ocorre devido à extrema sofisticação das heurísticas dos programas comerciais, que têm sido

⁶⁰ estimativas veiculadas informalmente e ocasionalmente, nos sites de fabricantes de programas de xadrez (por exemplo www.rwbel.nl e www.chessbase.com), bem como no fórum de discussão “Computer Chess Club” (disponível para membros em www.icdchess.com/cccl.htm).

⁶¹ Conhecimento é a palavra usada na literatura de Inteligência Artificial, muitas vezes com uma certa ingenuidade, para indicar a tradução aproximada do saber humano para um conjunto de regras e procedimentos de cálculo programados em computador. A tradução do conhecimento humano em procedimentos e regras é, por si só, um problema dos mais relevantes, que nos remete à questão das heurísticas, discutida mais adiante, neste capítulo.

⁶² Curiosamente, os mais rápidos programas da atualidade, rodando em microcomputadores de última geração, calculam mais de um milhão de posições por segundo, a um custo inferior a US\$2.500,00. Esse é um desempenho superior ao alcançado pelo do Deep Thought por volta de 1990, a um custo superior a US\$50.000 da época.

⁶³ Essa partida histórica encontra-se reproduzida no Apêndice II.

refinadas durante décadas, para fazer face ao poder de processamento relativamente reduzido dos computadores pessoais.

O incremento do poder computacional bruto proporcionou, durante muito tempo, um ganho de performance notável dos programas de xadrez, em relação aos melhores jogadores humanos. Dessa forma, os jogadores medianos, assim como a maioria dos mestres internacionais, foram convincentemente superados pelos melhores programas, nos últimos anos. Entretanto, uma vez alcançado o nível dos grandes mestres de topo, os programadores estão tendo problemas em demonstrar a superioridade de seus programas de modo conclusivo, apesar dos contínuos aprimoramentos do hardware e do software.

Os raros revezes dos computadores, em jogos contra grandes mestres, têm forçado os programadores a reconsiderar o peso relativo das heurísticas e da velocidade de busca como fatores de sucesso dos programas de xadrez. Até poucos anos atrás, a velocidade de busca era tida como *o fator* determinante da força dos programas. Hoje em dia, porém, já se admite que o desenvolvimento de heurísticas mais sofisticadas é uma condição necessária para que os programas superem certas dificuldades enfrentadas nas partidas contra os jogadores de primeira linha. Em grande medida, essa tendência resulta da popularização do xadrez computacional, que está fazendo com que os jogadores humanos aprendam a explorar as deficiências características do jogo das máquinas, conduzindo as partidas por vias em que a força bruta tem menos valor do que a compreensão dos fatores posicionais.

Ainda é cedo para se verificar se o crescimento do rating performance das máquinas está se desacelerando, em competições contra seres humanos. Contudo, ao se analisar as últimas competições em que tomaram parte seres humanos e computadores, tem-se a impressão de que as máquinas obtêm, frente a jogadores de segunda linha, um sucesso desproporcional, ou seja, o rating performance das máquinas diante de jogadores mais fracos parece ser significativamente maior do que o alcançado frente a jogadores de topo.

Comparando dois artigos da equipe responsável pelo Deep Blue (anteriormente, Deep Thought), encontramos um forte indício de que o *conhecimento embutido* está se aproximando em importância da força bruta, como fator de desempenho dos programas frente a jogadores de primeira linha. Há um pouco mais de dez anos Hsu e al. (1990) afirmavam que:

Nas partidas jogadas após agosto de 1988, quando o computador atingiu a sua *velocidade analítica* atual, de 750.000 posições por segundo, seu rating performance superou a marca de 2600.

A próxima geração da máquina, que deve jogar sua primeira partida ao longo de 1992, rodará em um hardware muito mais poderoso. O equipamento aumentará a velocidade de análise em mais de 1.000 vezes, para cerca de um bilhão de posições por segundo. *Esta mudança, por si só, provavelmente poderia fazer do descendente do Deep Thought um jogador de xadrez mais forte do que Kasparov – ou do que qualquer outro ser humano na história* (p.18, grifos meus).⁶⁴

Há neste trecho, dois pontos marcantes: primeiro, a identificação entre a “capacidade analítica” da máquina e sua velocidade de cálculo, o que mostra claramente a importância atribuída pelos autores à força bruta. Segundo, a convicção de que o aumento puro e simples da capacidade de cálculo faria do sucessor do Deep Thought o jogador mais forte de todos os tempos (“Esta mudança, por si só...”). Porém, com o passar dos anos, o investimento em velocidade de cálculo parece ter deixado de ser tão prioritário, pois a máquina que enfrentou Kasparov, já em 1997, era *apenas* 200 a 300 vezes mais rápida do que a de 1989, e não mil vezes, como estava proposto para a máquina de 1992. A esse respeito, são reveladores os comentários de Campbell e al. (2002), sobre os esforços realizados entre os matches de 1996 e 1997:

Depois do match de 1996 contra Kasparov, ficou claro que havia uma série de deficiências no jogo do Deep Blue I. Uma série de mudanças foram feitas durante a preparação para a revanche que aconteceu em maio de 1997. Primeiramente, foi projetado um novo circuito integrado de xadrez⁶⁵, significativamente aprimorado. O novo circuito integrado tinha uma função de avaliação completamente redesenhada, indo de aproximadamente 6.400 parâmetros para mais de 8.000. [...] *Finalmente, nós concluímos que a capacidade de busca do Deep Blue era aceitável, e gastamos a maior parte*

⁶⁴ In the games played after August of 1988, when the computer reached its current *analytical speed* of 750,000 positions per second, its performance rating exceeded 2600.

The next generation of the machine, expected to play its first game some time in 1992, will run on far more powerful hardware. The equipment will increase the speed of analysis by more than 1,000-fold, to about a billion positions per second. *This change alone* might well make Deep Thought’s descendant a stronger chess player than Kasparov—or any other human being in history (p.18, grifos meus).

⁶⁵ Isto é, um dispositivo semicondutor especialmente projetado para cálculos relacionados com a avaliação das posições do xadrez.

*do nosso tempo entre os dois matches redesenhando, testando e refinando a função de avaliação (p.59, grifo meu).*⁶⁶

Portanto, a equipe do Deep Blue optou por trabalhar prioritariamente sobre as heurísticas da máquina, por considerar que este era o seu ponto fraco, e que, ao contrário, a velocidade de cálculo já era aceitável. Essa decisão está em flagrante contraste com a crença anterior, de que o incremento da capacidade de cálculo, *apenas*, seria suficiente para superar o Campeão Mundial.

A mudança de ênfase, do aumento da velocidade de cálculo de variantes, para o refinamento das heurísticas do Deep Blue, evidencia as dificuldades enfrentadas na tentativa de se superar as habilidades analíticas do ser humano, com base no poder computacional bruto. Concretamente, os evidentes desníveis observados no jogo do Deep Blue e de seus antecessores lança aos engenheiros e programadores o desafio de construir programas mais *capazes*, e não somente máquinas mais poderosas. Esse desafio é confirmado pelos resultados, e sobretudo pelo desenvolvimento dos jogos entre o Deep Blue e os grandes mestres que tiveram a oportunidade de testá-lo⁶⁷.

⁶⁶ After the 1996 match with Kasparov, it was clear that there were a number of deficiencies in the play of Deep Blue I. A series of changes were made in preparation for the rematch which took place in May of 1997. First, a new, significantly enhanced, chess chip was designed. The new chess chip had a completely redesigned evaluation function, going from around 6400 features to over 8000. [...] *Finally, we concluded that the searching ability of Deep Blue was acceptable, and we spent the vast majority of our time between the two matches redesigning, testing and tuning the evaluation function*, (p.59, grifo meu).

⁶⁷ O Deep Blue é um modelo de sucesso em Engenharia, entendida como um campo de trabalho que se utiliza da tecnologia para alcançar objetivos concretos, com reflexos sensíveis no mundo físico e/ou no mundo social. O objetivo do projeto Deep Blue foi claro, desde o começo: derrotar o campeão mundial de xadrez em um match, sob condições de torneio. Como vimos, esse objetivo foi alcançado em 1997, após alguns milhões de dólares investidos e uma década de trabalho intensivo. Portanto, o redirecionamento da prioridade do projeto, da força bruta para heurísticas mais elaboradas, tem um significado especial, porque não se deve à intenção de imitar o pensamento humano, mas simplesmente à necessidade de tornar o Deep Blue mais eficaz. O Deep Blue II tinha 30GB de memória RAM, e 120GB em unidades de disco, além de 30 microprocessadores de uso geral e 480 processadores específicos para otimizar as buscas no jogo de xadrez, trabalhando em paralelo. Nesse contexto, temos um testemunho eloquente das dificuldades de se superar os seres humanos em tarefas analíticas, através do poder computacional bruto. Que um ser humano tenha ganho uma partida, e empatado três, contra essa máquina, é um feito que merece explicação.

Dois dos maiores programadores de xadrez da atualidade, Amir Ban e Mark Uniacke, em declarações recentes, reforçam a idéia de que o refinamento das heurísticas está sendo considerado cada vez mais importante, na medida em que os ganhos com o poder computacional bruto das máquinas não têm sido suficientes para derrotar os maiores jogadores humanos. Amir Ban, de Israel, é responsável, juntamente com seu colega Shay Bushinsky, pelo programa Junior, um dos mais fortes programas comerciais de todos os tempos. Recentemente, perguntado sobre quais as mudanças mais importantes entre as duas últimas versões do seu programa, Amir Ban respondeu:

Avaliação. Foi esse o foco do meu trabalho no Junior 6, e no Junior 7 eu continuei dessa forma e conduzi o trabalho por vias muito interessantes. Eu tenho pensado há muito tempo no que queremos dizer quando falamos que avaliação de uma posição é essa ou aquela, e quando afirmamos que um programa avalia melhor do que o outro (Ban, 2002).⁶⁸

Mark Uniacke, criador do programa Hiarc, sempre foi um dos partidários da tese de que o refinamento das heurísticas é o ponto crucial para melhorar o desempenho dos programas, em um nível competitivo elevado. Eis o que ele afirma a respeito do Hiarc versão 8, que passou aproximadamente três anos em desenvolvimento:

Para o Hiarc 8, eu alterei significativamente a forma e a profundidade da árvore de busca, sendo mais seletivo no que tange às variantes exploradas[...] Alguns aprimoramentos foram feitos em termos da velocidade bruta de processamento, totalizando cerca de 15%, *mas a maior parte desse ganho foi revertido nas funções de avaliação e na inteligência da busca*[...] No Hiarc 8 eu reescrevi o código de avaliação de estruturas de peões e o programa agora *entende* muito mais sobre quando os peões estão fracos, incluindo debilidades potenciais e como o controle das casas e as

⁶⁸ “Evaluation. It was the focus of my work on Junior 6, and in Junior 7 I continued this and took it to very interesting directions. I have been thinking for a long time what we mean when we say the evaluation of a position is this or that, and what we mean when we say that one program evaluates better than another (Ban, 2002)”.

peças influenciam estruturas de peões e vice-versa⁶⁹ (Uniacke, 2002, grifos meus).

Observa-se, assim, que mesmo os aprimoramentos nas rotinas de busca foram feitos para torná-las mais seletivas, isto é, para reduzir (ao invés de aumentar) o número de lances considerados para avaliação, de modo que as variantes possam ser calculadas mais a fundo. Essa técnica está em flagrante contraste com o simples aumento do número de variantes calculadas, que seria a aplicação da força bruta em sua forma mais elementar. Além disso, chama a atenção o fato de Uniacke ter aproveitado a otimização da rotina de busca para introduzir mais *conhecimento* nas rotinas de avaliação (heurísticas), de modo que o programa, nas suas palavras “entenda” melhor certos traços sutis da estratégia enxadrística, como as estruturas de peões. Em partidas recentes do Hiarc 8, contra dois grandes mestres consagrados (Apêndice III), o programa obteve uma vitória e três empates, indicando que a abordagem de Uniacke resultou em um programa que joga um xadrez muito consistente, do ponto de vista posicional.

A completa superação do homem pela máquina, no xadrez, ainda não aconteceu, malgrado os enormes avanços do hardware e, de um modo geral, as máquinas continuam apresentando um desempenho bastante desigual, alternando partidas notáveis com outras, de baixo padrão, que não agradariam nem mesmo a enxadristas amadores de força mediana.

Grandes mestres humanos também cometem erros crassos, geralmente em função de fatores psicológicos, ou mesmo devido a um mau preparo físico. Os erros humanos ocorrem também, em função da dificuldade em se calcular longas variantes, em posições muito complicadas. As circunstâncias em que os erros das máquinas são cometidos, por outro lado, revelam um problema epistemológico, que se manifesta no descolamento entre o alto padrão de jogo, alcançado na maioria das vezes, e a impossibilidade das máquinas de *entender* certas nuances das posições de xadrez. Os cálculos das máquinas são, quase sempre, exatos, e sua capacidade tática em posições complexas é assombrosa. Porém, apesar dos grandes progressos do xadrez computacional, a avaliação posicional dos programas ainda deixa muito a desejar, e sua

⁶⁹ For Hiarc 8, I significantly changed the shape and depth of the search tree by being more selective about the variations explored[...] Some raw speed improvements were made totalling about 15%, but *most of this was ploughed back into the evaluation functions and search intelligence*[...] In Hiarc 8 I rewrote the pawn evaluation code and the program now *understands* much more about when pawns are weak including potential future weaknesses and how the square control and pieces influence pawn structures and vice-versa.

incapacidade de fazer planos concretos freqüentemente provoca lances inconsistentes, em situações em que não há variantes forçadas, onde a força bruta é privilegiada⁷⁰.

4.3. Problemas de desempenho

As derrotas das máquinas frente a jogadores humanos são fruto de problemas conhecidos, que eventualmente se manifestam em padrões de jogo ineficazes. Em linhas gerais, os programas de xadrez têm dois problemas que afetam negativamente o seu desempenho: (1) O chamado “efeito-horizonte”, relacionado à impossibilidade dos programas de perceber ameaças situadas além da profundidade máxima da sua árvore de busca, e (2) O problema das heurísticas, relacionado com a transposição imperfeita do conhecimento enxadrístico do ser humano para as funções de avaliação usadas pelos programas.

O efeito-horizonte ocorre quando o computador calcula uma seqüência de lances, até determinada profundidade, mas deixa de fora dos seus cálculos lances que estão um pouco além, com reflexos decisivos no desenvolvimento da partida. Para exemplificar o efeito-horizonte, de forma simples, suponhamos que, em determinada posição, cabe ao programa jogar e, após calcular algumas variantes até a décima quarta meia-jogada, ele opta pelo lance A, porque seus cálculos indicam que as melhores respostas, digamos, A₁, A₂ e A₃, deixarão o adversário em posição ligeiramente inferior, enquanto a resposta A₄ deixará o adversário com uma dama⁷¹ a menos, o que em geral se constitui em uma desvantagem material decisiva. Mas, suponhamos que o adversário da máquina (um humano de primeira linha) tenha calculado as variantes críticas até a vigésima meia-jogada, chegando à conclusão que, com o lance A₄, entregará a dama, mas dará xeque-mate de maneira forçada, três lances depois ou, se a dama não for capturada, o computador ficará em posição inferior. Dir-se-ia, em sentido figurado, que o lance A₄, feito pelo humano, pegou o computador *de surpresa*, porque suas conseqüências reais estavam *escondidas* para além do horizonte de cálculo da máquina, quando o lance A foi escolhido. Em resumo, o lance A, desfavorável ao programa, foi avaliado como favorável, devido ao efeito horizonte.

⁷⁰ Alguns aspectos técnicos das limitações dos programas de xadrez são discutidos na próxima seção, e vários casos concretos em que tais limitações se evidenciam são analisadas no Apêndice I. Ver também o exemplo de avaliação posicional distorcida exposto na seção 4.5.

⁷¹ Peça também chamada de “rainha”. Dama é o nome adotado pela comunidade enxadrística brasileira, e também nos livros de xadrez editados no Brasil.

Naturalmente, uma das formas de se minimizar o efeito horizonte é o incremento da velocidade de busca, através do aprimoramento do hardware e do software, visando calcular mais lances à frente. Outra abordagem é a usada por Mark Uniacke, conforme descrito na seção anterior, ou seja, tornar a busca mais seletiva, e conseqüentemente mais profunda, (estreitamento da árvore de busca). Nesse caso, corre-se o risco de que variantes importantes fiquem de fora da busca, mas esse risco é adequadamente compensado pelo ganho de performance obtido, devido ao aprofundamento na análise das variantes críticas.

Contudo, nas competições de alto nível, a seletividade, o incremento do poder computacional e o aumento da eficiência das rotinas de busca são insuficientes para minimizar o efeito-horizonte a um nível satisfatório. Para superar essa dificuldade, os programadores introduzem técnicas que permitem aos programas reconhecer as variantes mais críticas, ou seja, aquelas em que há maior risco de respostas inesperadas por parte do adversário. Tipicamente, os programas são elaborados de modo a calcular mais a fundo as variantes em que ocorrem posições instáveis, ou seja, posições que podem ter o balanço material ou posicional alterado abruptamente. Dentre os indicadores mais comuns de que uma posição é instável, podem ser citados: (1) um dos reis está em xeque, (2) uma peça acabou de ser tomada e (3) há um ou mais peões avançados à sexta ou à sétima linha (Levy e Newborn, 1991, p.174). Calculando mais a fundo as variantes em que ocorrem posições com essas características, os programas minimizam o efeito-horizonte, muito embora não o eliminem por completo. Quando há tempo disponível, os bons programas aprofundam o cálculo das variantes críticas até atingir posições que são estáveis, segundo seus critérios de avaliação.

É desejável o aprofundamento da árvore de busca naquelas linhas em que a avaliação posicional sofre mudanças bruscas, entre um lance e outro. Por isso, os programas geralmente calculam de modo aprofundado as principais variantes das posições abertas, com jogo de peças intenso e ameaças táticas concretas. Em contrapartida, calculam de forma mais superficial as variantes das posições fechadas, sem ameaças concretas, com menor mobilidade das peças⁷². Essa é uma das razões pelas quais até mesmo os melhores enxadristas, atualmente, preferem as posições fechadas, nas suas partidas contra computadores: pois é justamente nas posições abertas, em que os programas

⁷² A classificação das posições em “fechadas” e “abertas” não é absoluta, mesmo porque a transição entre posições de ambos os tipos pode se dar em poucos lances. Além disso, há posições fechadas em que ameaças táticas concretas, ou seqüências de lances forçadas têm impacto sobre o desenvolvimento do jogo. Ressalvadas as exceções, contudo, vale a regra de que os programas calculam melhor em posições abertas.

manifestam todo o seu potencial, que a visão combinatória humana está mais sujeita a falhas⁷³. O efeito horizonte, porém, persiste, ainda que de forma muito mais sutil do que no passado. Ainda hoje, as máquinas são eventualmente surpreendidas taticamente por jogadores humanos de primeira linha, por não ter calculado certas variantes críticas com a profundidade suficiente.

A questão das heurísticas será analisada, em seus desdobramentos epistemológicos, na próxima seção do presente trabalho. Por enquanto, basta lembrar que as heurísticas são implementadas através das funções de avaliação posicional, usadas pelos programas para limitar a largura e a profundidade da árvore de busca, conforme explicado na seção 3.5. A escolha de um lance depende diretamente das funções de avaliação, exceto nos casos raros em que a máquina pode calcular todo o desdobramento do jogo, até um xeque-mate ou um empate forçado.

O algoritmo mini-max avalia o mérito relativo de cada variante com base na avaliação da posição analisada no nível mais profundo da árvore de busca. Conseqüentemente, quanto mais precisa for a avaliação posicional, maiores são as chances de se escolher um bom lance. Mas, se a avaliação posicional tiver imperfeições, há a possibilidade de se escolher mal uma variante, porque detalhes posicionais para melhor ou para pior não serão corretamente ponderados. Além disso, as funções de avaliação também influenciam a seleção das variantes que serão analisadas mais a fundo. Portanto, funções heurísticas deficientes podem fazer com que o programa ignore variantes promissoras, tanto para o próprio programa, quanto para o adversário. Por exemplo, o programa pode analisar uma jogada do adversário, no decorrer de uma variante, e concluir que não há maiores riscos, quando na verdade, um fator posicional sutil, ausente das ponderações

⁷³ Na comunidade interessada em xadrez computacional há quase um consenso em torno do que se convencionou chamar de jogo “anti-computador”, ou seja, a adaptação do estilo humano para jogar contra as máquinas. Na verdade, os enxadristas sempre trabalharam com fatores psicológicos, tentando conduzir as partidas por linhas mais agradáveis para si próprios e desagradáveis para seus adversários, em função de fatores como estilo de jogo, experiência prévia com essa ou aquela abertura e posicionamento nos torneios. Agora, com o advento dos programas de ponta, a adaptação de estilo se desloca do campo psicológico para o campo epistemológico, porque o desafio dos grandes mestres é conduzir as partidas por vias que favoreçam seu conhecimento das sutilzas do jogo, em oposição ao enorme poder de cálculo e as heurísticas comparativamente ingênuas das máquinas. Recentemente, Loek van Wely (REBEL,2002), um dos vinte enxadristas mais fortes do mundo, afirmou que “Você não pode jogar no seu próprio estilo contra o computador, isso é praticamente suicídio, você tem que adotar uma estratégia anti-computador e estar bem preparado” (“You cannot play your own style against the computer, that's about close to suicide, you must play the anti-computer strategy and be well prepared”).

da função de avaliação utilizada, faz com que a posição do programa fique comprometida. Conseqüentemente, uma avaliação posicional imperfeita prejudica o programa naquilo que ele tem de melhor, que é a capacidade de calcular um grande número de variantes relevantes para o tratamento de determinada posição de jogo.

É importante ressaltar que as heurísticas usadas pelos melhores programas são sofisticadíssimas, e incorporam mais *conhecimento enxadrístico* do que o evidenciado pela maioria dos jogadores amadores de alto nível. Portanto, as imperfeições das funções de avaliação, a que se refere o parágrafo anterior, não são imperfeições grosseiras, mas problemas muito sutis, que geralmente só se manifestam em um padrão de jogo muito elevado.

Muitos dos empates dos computadores frente a humanos de força mediana são resultantes de posições sólidas e taticamente estéreis, onde o equilíbrio é difícil de romper, desde a abertura. Nesses casos, os jogadores de força mediana às vezes conseguem manter a posição sob controle, sem criar pontos fracos, e assim chegar ao empate. Já os jogadores de força superior buscam provocar debilidades posicionais no campo adversário, através de manobras sutis e de longo alcance. Os computadores, porém, não têm esta capacidade de planejamento, e na ausência de erros mais evidentes, por parte dos adversários, geralmente continuam fazendo lances bons e sólidos, porém inócuos. Eventualmente, fazem lances imprecisos, em desarmonia com o caráter da posição, dando aos grandes mestres a oportunidade de construir uma vantagem decisiva. Apesar de se defenderem extremamente bem, em posições delicadas, os programas têm se demonstrado incapazes de buscar lances estrategicamente refinados, para desequilibrar posições de aspecto estéril, e induzir reações inexatas, por parte dos oponentes⁷⁴.

Nos próximos parágrafos, serão discutidas algumas das manifestações típicas do efeito-horizonte e dos problemas com as funções de avaliação. Os pontos aqui discutidos são expressões típicas das limitações técnicas dos programas de xadrez, evidentes o bastante para chamar a atenção de enxadristas de nível mediano. Outras limitações há, que só se

⁷⁴ Nesse sentido, os grandes enxadristas não têm propriamente o dom de blefar, mas a capacidade de propor aos seus adversários, problemas difíceis de resolver.

revelam a jogadores de muito alto nível⁷⁵. Feita essa ressalva, vejamos algumas das situações em que os programas enfrentam dificuldades:

CRAVADA PERMANENTE⁷⁶: Ocorre quando uma peça fica cravada por tempo indeterminado, sem possibilidade de liberação, em condições adequadas. A peça cravada, em termos meramente formais, pode agir sobre várias casas do tabuleiro, e portanto, pode ser considerada como possuindo boa mobilidade. Isso contribui para falsear a avaliação posicional do programa, que é feita de modo estático. Um exemplo é o diagrama 1, abaixo, que reproduz a posição final de uma partida entre Anand e o programa Fritz 6, disputada em 1999, onde o bispo do programa se encontra irremediavelmente cravado, ocasionando um empate, apesar da vantagem material das Negras.

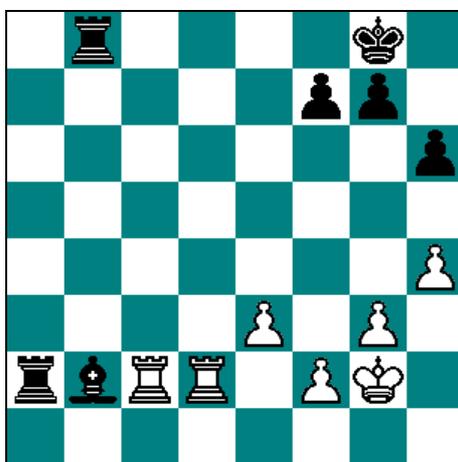


Diagrama 1: Cravada permanente

Como a peça cravada mantém o seu valor nominal inalterado, o programa não analisa a cravada como um problema real da posição, na ausência ameaças táticas detectáveis no seu horizonte de cálculo.

⁷⁵ Discuto as limitações dos programas, em termos epistemológicos e em termos técnicos, sem qualquer pretensão de fazer face a esses mesmos programas, na prática do jogo. De fato, assim como quase todos os humanos, praticamente não tenho chances de obter sequer um empate frente aos melhores programas da atualidade. Por outro lado, não é fácil encontrar referências explícitas às limitações dos programas, porque, como é compreensível, nem os programadores e nem os grandes mestres de primeira linha, envolvidos com os projetos de xadrez computacional, estão dispostos a divulgar seus segredos profissionais abertamente.

⁷⁶ No xadrez, uma peça cravada é aquela que não pode deixar a sua posição, sem expor uma peça de maior valor. Se a peça a ser exposta a uma ameaça for o rei, é ilegal mover a peça cravada. Caso contrário, é legal movê-la, mas geralmente, indesejável. Por exemplo, se um cavalo branco estiver diante da dama branca, interrompendo a linha de ataque de uma torre negra, podemos dizer que o cavalo está cravado, porque se ele se mover, a torre poderá tomar a dama.

BISPOS DE CORES OPOSTAS. É um problema que se insere em um contexto mais amplo, e um tanto sutil, que é a passagem do meio-jogo ao final. Geralmente, os finais com bispos de cores opostas tendem ao empate, se um dos lados não tiver uma vantagem material significativa. Contudo, se as damas e/ou as torres estiverem presentes no tabuleiro, o valor da iniciativa cresce muito, com a possibilidade de se explorar pontos fracos no campo adversário. O problema é que muitas vezes tais oportunidades de ataque estão além do horizonte de cálculo dos programas, que terminam por analisar as posições de modo distorcido.

POSIÇÕES FECHADAS, COM POSSIBILIDADES DE ATAQUE ATRAVÉS DE RUPTURAS. Mesmo os melhores programas ainda têm dificuldades para manter o seu padrão de jogo em posições fechadas, que requerem planos estratégicos de longo prazo. A rigor, os programas de xadrez não planejam. Ao invés disso, eles *procuram* por soluções *contingentes*, como se cada posição que ocorre durante uma partida fosse um novo problema a ser resolvido. Nas posições de bloqueio, com cadeias de peões fechadas, os bons jogadores humanos tentam reorganizar suas peças para efetuar (ou prevenir-se de) rupturas. Tal planejamento requer uma visão antecipada dos tipos de posição que podem se seguir, bem como o conhecimento das relações entre as peças e as estruturas de peões. Os programas, na ausência de longas variantes forçadas, freqüentemente fazem movimentos em desarmonia com as nuances sutis do posicionamento das peças, e em função disso sucumbem a ataques montados lentamente, por detrás dos peões avançados. Recentemente o excelente programa Shredder 6 perdeu uma partida jogada contra o grande mestre Smirin, em apenas 24 lances, porque falhou em antecipar um ataque decisivo, disfarçado sob uma posição fechada (ver partida 6, no Apêndice I).

TRANSFORMAÇÕES POSICIONAIS SÚBITAS, ATRAVÉS DE SACRIFÍCIOS. Nem sempre os sacrifícios são analisados detalhadamente pelos computadores, porque às vezes as compensações obtidas são de ordem posicional, e seus efeitos concretos estão além do horizonte de cálculo. Assim, os jogadores humanos ainda vencem ou empatam boas partidas contra os computadores, utilizando-se de sacrifícios cujos efeitos escapam, a um só tempo, tanto às funções de avaliação quanto às possibilidades de cálculo dos programas. Tais transformações posicionais súbitas são possíveis porque o item de maior peso nas funções de avaliação, por larga margem, é a contagem das peças em jogo. Portanto, as variantes em que o adversário sacrifica material geram, a princípio, avaliações muito favoráveis à máquina. Nessas circunstâncias, se o adversário puder restabelecer o equilíbrio de material, ou obter um ataque decisivo ao rei, nos lances seguintes ao do sacrifício (dentro do horizonte de cálculo), o programa jogará lances preventivos, e não

permitirá o sacrifício. Caso contrário, o programa avaliará o sacrifício como uma troca de peças desvantajosa para o ser humano, e permitirá que o adversário obtenha vantagens posicionais que se materializarão a longo prazo.

TRANSIÇÃO ENTRE MEIO-JOGO E FINAIS DE PARTIDA. Os finais de partida – grosso modo caracterizados como situações em que restam poucas peças no tabuleiro, e/ou as damas foram trocadas – exigem um tratamento diferente daquele dispensado ao meio-jogo. Por um lado, o número de lances válidos por posição, nos finais é geralmente menor do que no meio-jogo. Por outro lado, as variantes são geralmente muito mais longas, a ponto de, muitas vezes, serem intratáveis através da força bruta. Além disso, as heurísticas para a avaliação das posições em finais de partida são muito diferentes das heurísticas para a avaliação de posições no meio-jogo⁷⁷. Porém, é difícil saber exatamente que heurísticas utilizar, pois a transição entre as fases do jogo não é rigorosamente delimitada. Que significa dizer que há *poucas* peças em jogo? Dez peças? Onze? Quando as damas saem do tabuleiro, entra-se automaticamente no final? Como essas perguntas não têm respostas exatas, os programas têm que usar heurísticas para a aplicação de heurísticas, isto é, regras que permitam reconhecer que funções de avaliação utilizar em determinadas posições. Os jogadores humanos, por sua vez, simplesmente *reconhecem* o tipo de posição com que se defrontam, sem recorrer a regras específicas. Além disso, os humanos são treinados para jogar os finais de partida com base em uma abordagem *teleológica*, à qual Shereshevsky (1985, p.55) dá o nome de “pensamento esquemático”. O método consiste em visualizar o tipo de posição que se deseja atingir, e depois elaborar as operações táticas pertinentes, tentando conduzir o jogo pelas linhas desejadas. Na maioria dos finais, esta abordagem é indispensável, pois o número de variantes a calcular é intratável para os seres humanos. Também aqui, a incapacidade dos computadores de planejar o jogo se faz notar. O ser humano pode imaginar posições futuras, a fim de escolher os caminhos mais promissores de jogo, sem recorrer ao cálculo exaustivo de variantes. Para os programas, porém, a única forma de avaliar uma posição futura é calculando uma variante que conduza até ela. Embora se referindo ao jogo entre adversários humanos, Shereshevsky exprime algo dessa diferença entre humanos e computadores, na abordagem dos finais de partida:

In the endgame, schematic thinking gives an experienced player the advantage over an opponent who may be superior to him in rapidity and depth of calculation but who relies mainly on this calculation (op. cit., p.55).

⁷⁷ Por exemplo, no meio-jogo, é quase sempre fundamental que o rei permaneça em segurança, protegido por peças e peões, longe da área central do tabuleiro. Nos finais, ao contrário, o uso do rei como peça de ataque e defesa é de grande importância.

Conforme já foi explicado, na seção anterior, as bibliotecas de finais permitem que os computadores joguem com perfeição em posições com até seis peças em jogo. Em outros casos, porém, as deficiências históricas das máquinas nos finais ainda se fazem notar, principalmente em função da sua incapacidade de fazer planos.

4.4. A questão das heurísticas

Elaborar uma heurística para avaliar posições de xadrez significa exprimir, através de uma linguagem formal, o que os mestres e os livros têm a dizer sobre os aspectos posicionais do jogo, ou ainda, formalizar os princípios e técnicas que os mestres aplicam ao jogar xadrez. Feita a catalogação dos princípios e técnicas, sob a forma de regras e fórmulas, é necessário implementar a função de avaliação de modo eficiente, em termos computacionais.

Para um enxadrista forte, o tabuleiro é um espaço criativo e um campo de argumentação. É também laboratório de ensaios e caderno de teoremas, cheio de provas incompletas. Sob esta perspectiva, uma posição do jogo de xadrez envolve uma larga gama de conceitos, padrões, técnicas e procedimentos, apreendidos, experimentados e transformados no decorrer de um prolongado processo de interação sócio-cultural. Cabe questionar até que ponto essas nuances do jogo podem ser traduzidas heurísticamente, mesmo considerando que o xadrez é um domínio eminentemente formal, sob uma perspectiva matemática.

Por outro lado, mesmo em termos puramente abstratos, uma posição de xadrez envolve incontáveis possibilidades de interação entre as peças do tabuleiro. A lógica interna do jogo manifesta-se através da sua sintaxe, que determina quais são os lances possíveis, de acordo com as regras, e também através de uma semântica, que constitui o campo dos lances significativos, em termos de uma boa condução das partidas. Contudo, as funções de avaliação usadas pelos programas de xadrez associam um único valor numérico a cada posição avaliada. Dessa forma, a apreciação das relações entre as peças em jogo é reduzida a um escalar, em um radical processo de abstração – sujeito a falhas, mas, na prática, bastante eficaz.

Tecnicamente, há três etapas fundamentais para a elaboração e sintonia de uma função de avaliação: (1) a definição dos aspectos posicionais a serem avaliados, que se manifestam através de padrões específicos de posicionamento das peças; (2) a criação

de um algoritmo para reconhecer estes padrões de interesse; e (3) o estabelecimento de escalas numéricas, ou “pesos” a serem introduzidos na função de avaliação.

Tomemos, como exemplo, os peões dobrados⁷⁸, que constituem um dos aspectos mais evidentes de uma avaliação posicional. Em geral, os peões dobrados são nocivos ao lado que os possui, principalmente se estiverem também isolados – quando seu efeito é realmente pernicioso. Por isso, as funções de avaliação dos programas pontuam negativamente os peões dobrados, e ainda mais negativamente os peões dobrados e isolados⁷⁹. Contudo, mesmo a introdução dessa heurística simples, não é trivial. Primeiramente, tem-se que fazer um algoritmo que rastreie todo o tabuleiro em busca de peões dobrados e isolados, de forma eficiente, para não comprometer a velocidade de execução do programa. Depois, é necessário pontuar negativamente (penalizar) os peões dobrados. Se a penalidade estabelecida for muito baixa, o programa tenderá a permitir que sua estrutura de peões seja prejudicada, sem compensações posicionais adequadas. Mas, se a penalidade for alta demais, o programa poderá subvalorizar outros aspectos importantes da posição, para evitar os peões dobrados e isolados apareçam no seu campo, ou para criá-los no campo adversário. Portanto, além de considerar cada fator posicional, isoladamente, as funções de avaliação têm que manter o equilíbrio na combinação entre os vários fatores, que pode ter efeitos inesperados⁸⁰.

Os peões dobrados, que geralmente se constituem em fatores posicionais críticos, são simplesmente insignificantes em certas oportunidades. Vejamos, por exemplo, a seguinte posição, apresentada por Mark Dvoretzky (1996, p.179):

⁷⁸ Peões dobrados são os peões de uma mesma cor que ocupam uma mesma vertical no tabuleiro. De modo geral, os peões dobrados são estrategicamente desvantajosos, porque têm pouca mobilidade, podem ser facilmente atacados e têm o seu avanço bloqueado por um conjunto numericamente inferior de peões adversários.

⁷⁹ Peões isolados são aqueles que não têm peões da mesma cor nas verticais adjacentes à sua, e portanto não podem ser apoiados por outros peões. Em geral, peões dobrados e isolados são um alvo de ataque fácil.

⁸⁰ Há por exemplo, certas posições em que os peões dobrados ajudam no controle das casas centrais do tabuleiro. Porém, a avaliação da força e da fraqueza relativa dos peões, nessas circunstâncias, não é trivial, e um jogador humano experiente fará seu julgamento com base em uma visão de conjunto, levando em conta as possibilidades de desenvolvimento do jogo. Se uma função heurística simplesmente avaliar os dois fatores posicionais em separado, provavelmente não dará resultados consistentes.

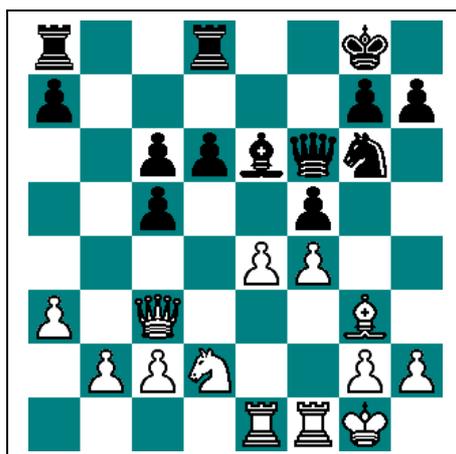


Diagrama 2: Valor relativo de uma debilidade

Aqui, as Brancas seguiram com o lance **22.e5**, na tentativa de enfraquecer ainda mais os peões dobrados das Negras, na coluna **c**. Dvoretsky critica esse lance, nos seguintes termos:

A fraqueza de **c5** não tem a menor importância. Muito mais importante para as Brancas é o aparecimento do horrível peão **e5**, que bloqueia o seu próprio bispo, cavalo (depois do lance **24.Cf3**) e torre⁸¹.

Entretanto, como as heurísticas lidam com os aspectos gerais da posição, de forma monolítica, o programa não pode, de forma alguma, considerar o peão de **c5** “sem a menor importância”. Na melhor das hipóteses, a função de avaliação irá ponderar convenientemente os diversos fatores posicionais, em conjunto, mas nunca analisar, em separado, dois fatores que se contrapõem, tal como fez Dvoretsky, a partir da sua larga experiência como jogador de torneios e professor de grandes enxadristas.

Na seção 4.3, discutiu-se a dificuldade de se delimitar com exatidão a passagem do meio-jogo para os finais, com vistas à utilização de heurísticas apropriadas a cada fase do jogo. Especificamente, foi levantado o problema do rei, que normalmente passa de peça resguardada a peça ativa, na medida em que outras peças são trocadas, e o tabuleiro se esvazia. A esse respeito, vale à pena analisar o que diz Campbell (2002, p.76), sobre a heurística do Deep Blue:

Posições com a maior parte das peças no tabuleiro podem conservar o valor pleno do parâmetro de segurança do rei, enquanto os finais de jogo terão os

⁸¹ “The weakness in **c5** does not have the slightest significance. Of much greater importance for White is the appearance of a dreadful **e5**-pawn, which holds up its own bishop, knight (after **24.Nf3**) and rook.”

valores próximos a zero. Isso encoraja o Deep Blue, por exemplo, a trocar peças em posições onde o seu rei está menos seguro do que o rei adversário.⁸²

Sem dúvida, este trecho descreve a implementação coerente de uma regra geral, que, não obstante, comprometerá a avaliação de posições extraordinárias, em que o rei deve ser ativado, mesmo com muitas peças em jogo ou, ao contrário, em que mesmo com poucas peças em jogo existe a possibilidade de que o rei seja apanhado em uma rede de mate, se for ativado antes do tempo. Além disso, a tendência de trocar peças, caso o rei do adversário pareça mais seguro, é só em parte coerente, porque, freqüentemente, um rei mais seguro está também associado a uma melhor estrutura de peões, favorável nos finais de partida. Nesse caso, o Deep Blue poderia provocar trocas justamente quando o adversário tivesse as melhores perspectivas para um final.

Tomemos agora um conceito como o de *casa fraca*. Em xadrez, uma casa fraca é aquela que não pode ser defendida por um peão, e que está sem apoio das peças, ou precariamente apoiada. Assim, é uma casa que se transforma em uma base potencial para o posicionamento de peças adversárias, visando operações hostis. Ora, evidentemente, este é um conceito que deve ser incorporado às heurísticas de avaliação posicional. Contudo, como qualquer conceito geral, este só adquire significado em situações concretas. Eventualmente, uma casa identificada como fraca, com base em princípios gerais, poderá se transformar em um ponto sem interesse dentro da configuração específica do jogo. Isso ocorrerá, por exemplo, se o adversário não tiver os meios materiais ou o tempo necessário para explorar a casa com proveito. Não obstante, se a idéia de casa fraca tiver sido incorporada à heurística, todas as casas identificadas como fracas terão impacto sobre os valores calculados pela função de avaliação, independente do papel dessas casas na situação concreta de jogo.

Outro aspecto de avaliação posicional, muito importante para os programas de xadrez, é a mobilidade das peças. Quanto mais movimentos as peças têm disponíveis, maiores as possibilidades de se empreender operações táticas e operações posicionais relevantes, de ataque ou de defesa. Porém, também este é um conceito geral, cuja aplicação a uma situação específica de jogo está sujeita a uma série de considerações adicionais. Com efeito, as peças de um dos jogadores podem ter maior mobilidade, mas, ao mesmo tempo, um posicionamento menos harmônico do que as do adversário. Como a

⁸² Positions with most of the pieces still on the board may retain the full king-safety values, while endgames will have the values scaled close to zero. This, for example, encourages Deep Blue to trade pieces in positions where its king is less safe than the opponent's king.

harmonia entre as peças é um conceito difícil de traduzir heurísticamente, o fator mobilidade, considerado isoladamente, também pode distorcer bastante a avaliação de uma posição – muito embora, na maioria dos casos, seja predominante, sobretudo considerando-se a forma de jogar própria dos computadores.

Como vemos, há muitas possibilidades de conflito entre as regras gerais, expressas nas heurísticas, e os aspectos concretos das posições do xadrez. Portanto, a execução das funções de avaliação incorre frequentemente em *perdas de tempo*, ao introduzir nos cálculos fatores posicionais irrelevantes. Em casos extremos, além do esforço computacional desperdiçado, a avaliação das posições é distorcida, fazendo com que o programa escolha lances em desacordo com as necessidades da partida. Amir Ban assim se exprime sobre este ponto:

Jogadas ruins são parte do processo que leva a boas jogadas. Você pode formular a sua [função de] avaliação de modo a jogar muitíssimo bem em certas posições, e em algumas posições ela usa os mesmos critérios e provoca falhas lamentáveis. Infelizmente, você não pode apontar o dedo e dizer “Não jogue isso!”. Você tem que entender porque [a avaliação] falhou, e como consertá-la sem estragar o jogo, de forma geral. Isso não é fácil. Algumas vezes você tem simplesmente que dar de ombros e tentar avançar por vias mais claras⁸³ (Ban, 2002).

Tal observação, vinda de um dos mais bem-sucedidos programadores de xadrez da atualidade, mostra o quanto é difícil refinar as heurísticas – baseadas em abstrações – para lidar com as incontáveis particularidades das posições do xadrez: as heurísticas são gerais, mas os *segredos* das posições de xadrez se escondem na sua especificidade, que só a percepção privilegiada dos grandes jogadores, ou o poder de cálculo dos computadores, têm podido revelar. Em outras palavras: as heurísticas se situam no abstrato, mas o jogo de xadrez se resolve no concreto. É por isso que, no xadrez computacional, as funções de avaliação refinadas são um recurso indispensável, mas insuficiente para substituir a força bruta.

⁸³ Bad moves are part of the deal of good moves. You can formulate your evaluation to play excellently in many positions, and in some positions it uses the same criteria and comes up with a miserable failure. Unfortunately you can't wave a finger and say 'Don't play that!'. You need to understand why it failed, and how to fix it without damaging overall play. This is not easy. Sometimes you just need to shrug and try to advance in directions that are clearer.

A implementação das heurísticas em computador exige a codificação exaustiva de todos os fatores posicionais considerados relevantes, que tomarão parte dos cálculos de avaliação posicional. Por isso, o processamento de uma função de avaliação requer a execução de milhares de operações aritméticas, para cada posição avaliada. O Deep Blue II, por exemplo, reconhecia cerca de 8000 “padrões” de interação entre as peças (Campbell e al., 2002, p.73), contra aproximadamente 6.400 padrões reconhecidos por seu antecessor, o Deep Blue I. Embora as funções de avaliação dos melhores programas comerciais não sejam discutidas publicamente, em detalhes, sabe-se que elas são mais sofisticadas do que as do Deep Blue, e por isso são ainda mais *pesadas* em termos computacionais.

Na medida em que as heurísticas vão sendo refinadas, aumenta o tempo necessário para executar as funções de avaliação, o que diminui o número de variantes que o programa é capaz de calcular. Como a força dos programas apóia-se, em grande medida, no cálculo de variantes, uma heurística muito sofisticada pode, paradoxalmente, baixar o nível de jogo da máquina. Uma forma de atenuar esse conflito entre sofisticação heurística e velocidade de cálculo é executar a função de avaliação por partes, conforme descrito por Campbell (2002, p.63). O Deep Blue, usava uma “função de avaliação rápida” e uma “função de avaliação lenta”: a primeira, levando em consideração fatores posicionais fáceis de calcular e com maior peso na avaliação final da posição (em particular, o número de peças presentes no tabuleiro, e seu posicionamento), e a outra, levando em consideração fatores posicionais mais sutis e difíceis de calcular, e com menor peso na avaliação final. Se a “avaliação rápida” der um resultado acima de um determinado valor, então a “avaliação lenta” pode ser dispensada, pois deixa de ser relevante, e consumiria tempo de processamento inutilmente. Assim, se em determinada posição, a máquina avalia que tem o equivalente a duas peças de vantagem, a avaliação de aspectos posicionais sutis não é realizada, pois perde o sentido⁸⁴.

Reconhecida a eficácia das heurísticas, bem como as suas limitações, a discussão precedente pode agora ser sumarizada em três pontos fundamentais:

- (1) Uma função heurística avalia cada posição estaticamente, de modo que as transformações posicionais só são avaliadas após o cálculo das variantes. É muito difícil incorporar, em uma heurística, aspectos da dinâmica do xadrez, ou seja, avaliar heurísticamente *o que pode acontecer* em uma posição, já que as transformações posicionais relevantes são inúmeras, e nem sempre se revelam

⁸⁴ Duas peças de vantagem é um exemplo extremo, meramente ilustrativo. No artigo citado, os autores do Deep Blue não discutem em detalhes os parâmetros das funções de avaliação.

através de padrões definidos. Os enxadristas treinados enxergam em uma posição não apenas o seu caráter presente, mas também suas possibilidades futuras, sem recorrer, necessariamente aos cálculos exaustivos de variantes.

- (2) Uma heurística sempre se baseia na generalização de certos critérios de avaliação posicional, a partir de experiências progressas. Contudo, toda generalização estará sempre sujeita à refutação, por via dos fatos concretos.
- (3) Há situações para as quais parece ser impossível a definição de heurísticas relevantes, especialmente quando se tem um número muito reduzido de peças no jogo. Nesses casos, as grandes bases de dados, associadas à força bruta, têm sido os únicos meios disponíveis para se resolver os problemas práticos que ocorrem no xadrez.

As heurísticas não são inerentes aos problemas, ou seja, um determinado problema não determina *uma só heurística* adequada para resolvê-lo. A avaliação das posições do xadrez admite diferentes abordagens, e abordagens bastante diferentes entre si podem conduzir a resultados práticos semelhantes. No âmbito humano, jogadores de força semelhante têm freqüentemente estilos muito diversos, derivados da forma como abordam os problemas do jogo. No que se refere ao xadrez computacional, um dos melhores programas da atualidade, o Tiger, do francês Christophe Théron, vem em duas versões: o Chess Tiger e o Gambit Tiger. O primeiro utiliza uma função de avaliação convencional, e o outro, uma função de avaliação que privilegia um jogo de ataque especulativo, inclusive à base de gambitos⁸⁵. De acordo com vários relatos de aficionados do xadrez computacional, apresentados no fórum Computer Chess Club (CCC), o conjunto dos resultados obtidos pelo Chess Tiger e pelo Gambit Tiger, contra outros programas, é bastante semelhante, malgrado as diferenças marcantes entre as funções de avaliação de ambas as versões do programa⁸⁶. Isso indica que tanto a heurística convencional, quanto a heurística “agressiva” são igualmente eficazes, embora por vias diversas.

⁸⁵ Gambito é uma linha de jogo em que a parte atacante sacrifica um ou mais peões (raramente uma peça), para alcançar um posicionamento ativo de suas próprias peças, abrir linhas de ataque e, ao mesmo tempo, prejudicar o desenvolvimento das peças adversárias. Os gambitos são especulativos, na medida em que não há retorno garantido do material sacrificado, e nem a certeza de que o ataque será decisivo. Se o ataque não prosperar, e o lado que se defendeu com sucesso mantiver a sua vantagem material, o jogador do gambito possivelmente perderá a partida.

⁸⁶ Embora reconhecendo que o caráter difuso e informal desses relatos enfraquece o presente argumento, é importante assinalar que alguns dos maiores programadores de xadrez participam ativamente das discussões no CCC, inclusive o próprio Christophe Théron.

Atualmente, os cinco ou seis melhores programas comerciais de xadrez têm uma força de jogo praticamente nivelada, embora os seus autores não compartilhem seus segredos, no que tange às funções de avaliação. O refinamento das heurísticas, portanto, é um processo muito individualizado, e é praticamente impossível que dois programadores, trabalhando independentemente, com diferentes consultores de xadrez e bibliografia enxadrística distinta, cheguem a funções de avaliação muito parecidas. Esse é mais um indício de que heurísticas diversas, implementadas por pessoas igualmente competentes, produzem resultados práticos parecidos, na avaliação das posições de xadrez.

As heurísticas são construções lógicas, *condicionadas*, mas não *determinadas*, pela natureza dos problemas que se deseja atacar. O aperfeiçoamento das heurísticas, conseqüentemente, é sempre um processo interativo, já que não existe um percurso lógico pré-definido, que possibilite a criação de uma heurística ideal para a solução de um dado problema.

A engenharia do conhecimento⁸⁷, ao traduzir em algoritmos certos aspectos da razão, deixa de fora os elementos mais voláteis e sutis da experiência humana, apreendidos na vida social⁸⁸. O ser humano pode exprimir, por meio de regras, parte do seu conhecimento e algumas determinantes da sua conduta. Contudo, isso não significa que as regras expressas sejam de fato aplicadas, quando as pessoas se conduzem dessa ou daquela maneira⁸⁹. Ainda que um grande enxadrista dedique muito tempo e energia à elaboração de novas heurísticas para programas de xadrez, nada garante que a competência desse jogador possa ser satisfatoriamente transferida para um computador, no mesmo sentido que poderia ser transferida para um pupilo talentoso⁹⁰.

Se, por um lado, os jogadores humanos de alto nível avaliam as posições de forma mais refinada do que as máquinas, é certo que nenhum grande-mestre seria capaz de enumerar, de chofre, os milhares de padrões de análise usados nas funções de avaliação dos programas. Nenhum jogador humano poderia empreender a avaliação posicional em bases quantitativas, a não ser que postulássemos um mecanismo cerebral inconsciente e altamente eficiente para cálculos aritméticos, o que não parece razoável. Ao contrário,

⁸⁷ Engenharia do conhecimento é o ramo da IA que trata especificamente da tradução do conhecimento humano especializado em representações formais, passíveis de implementação em computadores.

⁸⁸ Essa tese é amplamente explorada por Collins (1992) e Collins e Kusch (1998).

⁸⁹ Esta é uma tese de inspiração Wittgensteiniana, explorada, em diferentes nuances, por Ryle (1984), Searle (1992; 1995); Dreyfus (1992), Collins (1992) e Collins e Kusch (1998).

⁹⁰ Esta questão será retomada na seção 5.3, “Implicações pedagógicas”.

os jogadores humanos analisam as posições com base em padrões visuais e conceitos abstratos, usando sua experiência como base criativa para novas soluções⁹¹.

Os programadores de xadrez continuam elaborando heurísticas cada vez mais completas e eficazes. Ao mesmo tempo, os melhores enxadristas do mundo continuam resistindo à força dos melhores programas. Em meio a esse impasse, persiste um problema epistemológico: ou a avaliação das posições de xadrez não se resolve nas regras e cálculos, ou os meios encontrados até agora, para formalizar o conhecimento dos grandes jogadores, são insuficientes. Se a capacidade dos programas para calcular variantes fosse limitada, digamos, a um décimo dos níveis atuais, eles teriam poucas chances contra os jogadores de elite, como acontecia há cinco ou dez anos. No xadrez computacional, as heurísticas racionalizam o uso da força bruta, mas de modo algum a tornam dispensável.

4.5. Eficácia x competência

Gisnberg (1998) afirma que as máquinas são mais fortes nos jogos que podem ser abordados satisfatoriamente através de cálculos seriados, enquanto as pessoas levam vantagem nos jogos onde predomina o “reconhecimento de padrões” (supostamente, uma atividade de natureza “paralela”). No caso do xadrez, a performance parelha dos melhores humanos e dos melhores computadores seria explicada pelo equilíbrio entre o poder de cálculo e a capacidade de reconhecer para o estabelecimento de um alto padrão de jogo.

No xadrez, o termo “reconhecimento de padrões” não tem uma definição clara, porque uma mesma posição pode ser percebida de modo diverso por diferentes jogadores, revelando padrões radicalmente distintos. Talvez, ao olhar uma determinada posição, um jogador mediano enxergue apenas um peão central isolado (um padrão), ao passo em que um grande mestre enxerga todo um complexo de casas fracas (outro padrão). Também é possível que um jogador enxergue uma mesma posição de formas diferentes, quando defrontado com adversários distintos. Na verdade, falar em *reconhecimento* de padrões, isoladamente, traduz muito pouco da atividade mental de um grande enxadrista. Os maiores jogadores não apenas reconhecem padrões, como também *revelam* a existência de padrões nunca antes observados.

⁹¹ A expansão desse tópico nos conduziria para além do escopo desse trabalho. Os aspectos cognitivos da maestria no jogo de xadrez são discutidos, por exemplo, por Puchkin (1976) e Krogius (1974).

No estudo e na prática do xadrez, o posicionamento das peças está permeado por conceitos complexos, tais como ataque, defesa e iniciativa, compartilhados no interior da comunidade enxadrística. Esses conceitos, cujo domínio tem reflexo decisivo sobre a qualidade técnica do jogador, não guardam relação necessária com a disposição das peças no tabuleiro. Uma determinada disposição de peças pode representar, para um jogador, um chamado à *defesa passiva*, enquanto para outro jogador de mesmo nível, e estilo diferente, representa um chamado à *defesa ativa*. Em termos lógicos, ambos podem estar certos, e tanto a defesa ativa quanto a defesa passiva serem igualmente eficazes na posição em foco.

Ao invés de tentar reduzir a aptidão humana para certos jogos ao “reconhecimento de padrões”, seria mais apropriado apreciar o desempenho relativo de homens e máquinas da seguinte maneira: as máquinas levam vantagem em jogos cuja estrutura favorece o poder de cálculo, enquanto os seres humanos se destacam nos jogos cuja estrutura favorece o julgamento qualitativo e a percepção de conjunto.

No xadrez, o equilíbrio entre os melhores jogadores e os melhores computadores persiste porque ambos, homens e máquinas, obtêm resultados semelhantes por caminhos radicalmente diferentes. Os humanos não têm a mesma capacidade de cálculo e a mesma precisão das máquinas. Além do mais, a memória dos jogadores só pode *decorar* uma modesta fração das posições de aberturas e finais que os computadores armazenam. Às máquinas, entretanto, falta a capacidade analítica dos grandes mestres, que as heurísticas não conseguem imitar. Talvez, o contínuo aumento do poder de cálculo dos computadores, aliado a heurísticas cada vez melhores, faça com que os programas superem definitivamente os grandes mestres de primeira linha, nos próximos anos, em partidas de competição⁹². Contudo, isso não significaria uma superação da capacidade humana pela força das máquinas, em sentido amplo, mas tão somente uma superação contingente, em um domínio favorável ao poder de cálculo dos computadores.

A capacidade de certos programas para simular fenômenos complexos, tais como os fenômenos climáticos, não implica que os computadores tenham a capacidade de compreender os fenômenos simulados. Analogamente, a eficiência dos programas de xadrez não é, necessariamente, uma demonstração de *inteligência enxadrística*. É

⁹² Lembrando que, no xadrez, superar quer dizer obter resultados consistentemente melhores, ao cabo de muitas partidas, disputadas contra adversários variados, em condições de torneio. Oficialmente, Kasparov não é mais o campeão do mundo, mas é o melhor enxadrista do mundo, porque seu rating é o maior, devido à consistência dos seus resultados nos torneios.

natural que os seres humanos não possam competir com os programas de simulação, fazendo bilhões de operações aritméticas com lápis e papel. Da mesma maneira, os enxadristas não podem competir com o poder das máquinas para o cálculo e “memorização” de variantes. Contudo, tanto nas simulações como no xadrez, é o ser humano que atribui significado aos cálculos que os computadores realizam.

Radicalizando a discussão sobre este ponto, Collins (1992, p.80) afirma que

[...] seremos forçados a admitir que nem as calculadoras de bolso, nem as unidades centrais [de processamento] podem realmente fazer aritmética, mas que elas podem apenas se ocupar de uma pequena parte mecânica que se encontra no centro da aritmética. São os usuários humanos de computadores que fazem o resto. Forçosamente, são os homens que executam todos os atos aritméticos que são logicamente anteriores e logicamente posteriores à parte mecânica.⁹³

De acordo com este raciocínio, os computadores de xadrez na realidade *não jogam xadrez*, porque o xadrez não se esgota no cálculo de variantes, e as funções heurísticas não são mais do que um *simulacro aritmético* para uma parcela modesta do raciocínio dos grandes jogadores. Parafraseando o trecho supracitado, podemos dizer que, ao determinar suas jogadas, os programas de xadrez deixam de fora os atos que são “logicamente anteriores e logicamente posteriores” à escolha de um lance.

Penrose (1994, pp.45-47) discute um exemplo extremo de como a eficácia competitiva dos computadores de xadrez não implica em um real entendimento do jogo. Na posição abaixo, as Negras têm uma enorme vantagem material, mas as Brancas podem evitar a derrota, simplesmente movendo o rei por detrás da sua impenetrável barreira de peões.

⁹³ [...] on sera forcé d’admettre que ni des calculatrices de poche ni des unités centrales ne peuvent réellement faire de l’arithmétique, mais qu’elles ne peuvent se charger que d’une petite partie mécanique que se trouve être au centre de l’arithmétique. Ce sont les utilisateurs humains des ordinateurs qui font le rest. Forcément, ce sont les hommes qui effectuent tous les actes arithmétiques qui sont logiquement antérieurs e postérieurs à la partie mécanique.

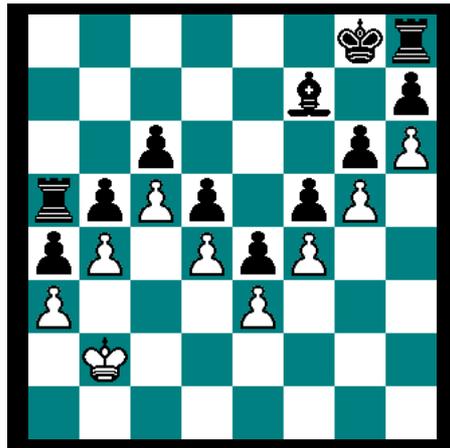


Diagrama 3: Fácil para os humanos, difícil para os computadores

Contudo, quando este problema foi “apresentado” ao Deep Thought, em 1993, ele capturou a torre da casa **a5**. Com isso, a barreira de peões se rompe, e as Brancas perdem o jogo. Assim, o que seria óbvio para qualquer jogador humano de fim-de-semana, não ficou evidente para uma máquina que já havia derrotado vários grandes mestres em jogos de competição. Mais interessante do que o resultado do experimento é o fato de que, antes de tomar a torre, o Deep Thought calculou milhões de variantes – demonstrando que, realmente, o programa é alheio a certas etapas lógicas que precedem a escolha de um lance⁹⁴.

A este exemplo, caberiam duas contestações: (1) o Deep Thought encontraria a solução se calculasse mais lances à frente, e (2) faltou apenas uma heurística adequada a este *tipo de posição*, para fazer com que a máquina seguisse o caminho correto. O argumento (1), com efeito, é revelador da fraqueza relativa da máquina, que teria que executar tantos cálculos para resolver um problema cuja lógica interna é tão clara para os seres humanos. O argumento (2), por outro lado, falha em definir o que é um “tipo de posição”. Certamente, poderíamos criar regras para definir que uma cadeia de peões totalmente bloqueada, e que não possa ser rompida pelo ataque das peças, leva ao empate. Na prática, muitas regras têm sido incluídas nas heurísticas dos melhores programas, para definir posições empatadas. Contudo, não parece razoável esperar que todas as situações inusitadas do jogo de xadrez possam ser previstas e tratadas por meio de regras específicas, tendo em vista a enorme quantidade de posições, com

⁹⁴ Realizei o mesmo teste com o Hiarc8, um dos mais poderosos programas de xadrez para microcomputadores, de indiscutível força competitiva. Após dez minutos de análise, e depois de calcular milhões de variantes, o Hiarc8 continuava a apontar a captura da torre como o melhor lance (ou o menos ruim), embora avaliando a posição das Brancas como totalmente perdida.

características peculiares, a que uma partida pode chegar⁹⁵. Ao se deparar com o exemplo acima, um jogador humano não se ocupa com regras e padrões, mas simplesmente compreende que há um empate, mesmo que nunca tenha visto ou discutido qualquer posição semelhante. As funções de avaliação baseadas em heurísticas, ao contrário, só tratarão convenientemente as posições para as quais já se tenham definido regras válidas. No xadrez computacional, as máquinas apenas seguem as regras que os seres humanos descobrem e formalizam⁹⁶.

Os enxadristas humanos de primeira linha são jogadores competentes, ao passo em que os atuais computadores de xadrez são *simuladores enxadrísticos* altamente eficazes. A convergência entre o poder de cálculo e capacidade de armazenamento de dados, nos computadores de xadrez, produz grandes adversários para treinamento, mas não cria grandes comentaristas, nem grandes aprendizes, e muito menos grandes professores de xadrez. Ryle (1984, p.55) afirma que “O executor inteligente de uma tarefa opera criticamente, e o expectador inteligente, acompanha criticamente.”⁹⁷ Mas os computadores de xadrez não fazem uma coisa nem outra, porque não são, de fato, jogadores inteligentes.

Por outro lado, nem os profissionais do xadrez, nem os amadores sérios podem dispensar, atualmente, o concurso dos programas de xadrez, nas suas atividades de treinamento. Os engenheiros e arquitetos também não podem dispensar o concurso dos computadores, e nem por isso os programas para cálculos estruturais ou projetos arquitetônicos são considerados inteligentes – ou competentes – nas suas respectivas especialidades. São as pessoas por detrás das máquinas que dão a elas o seu valor. Portanto, embora os computadores de xadrez não sejam em si mesmos competentes, sua força resulta da competência de muitas pessoas, *qualificadas e apaixonadas*, que durante décadas têm colaborado, direta ou indiretamente, para o desenvolvimento do

⁹⁵ Dependendo de como se interprete a questão, o número de “tipos” distintos de posições, no xadrez, é virtualmente ilimitado. Na prática, há incontáveis modos de classificar as posições, de modo que qualquer classificação sempre deixará de fora detalhes importantes, que outra classificação talvez pudesse capturar. No limite, a fim de capturar todos os detalhes essenciais de cada posição do xadrez, o número de “classes” seria igual ao número de posições possíveis.

⁹⁶ É tentador dizer, em tom provocativo: “Dê-me mais uma regra, e eu encontrarei dez exceções.” Em termos práticos, devemos lembrar que, quanto mais regras se inserem nas funções de avaliação, mais lentas elas se tornam, o que, a partir de um determinado momento, prejudica a qualidade de jogo do programa. Além disso, o aumento do número de regras faz crescer o tamanho do programa, o que, em um limite extremo, inviabilizará sua implementação.

⁹⁷ “The intelligent performer operates critically, the intelligent expectator follows critically.”

xadrez computacional. Daí a equipe responsável pelo Deep Thought, ao comentar suas expectativas para o confronto com Kasparov, ter afirmado que “o resultado não vai revelar se a máquina pode pensar, mas se um esforço humano coletivo pode suplantar os maiores feitos dos seres humanos mais capazes⁹⁸ (Hsu et al., 1990, p.24)”.

Nas partidas entre computadores e grandes mestres, portanto, não há choque entre as competências de homens e máquinas. O que há é o embate entre a competência de alguns seres humanos, devotados ao xadrez e desassistidos de recursos artificiais, e a competência de outros, dedicados à computação e à engenharia, e equipados com máquinas eletrônicas de última geração.

⁹⁸ “the result will not reveal whether the machine can think, but rather whether collective human effort can outshine the best achievements of the ablest human beings.”

5. O Saber Algoritmizado

5.1. Um mundo de algoritmos

Estamos tão acostumados aos algoritmos que nem nos damos conta de sua ubiquidade, que se revela por dois ângulos distintos. Primeiro, há inúmeros algoritmos que executamos todos os dias, para levar a cabo nossas atividades lúdicas e profissionais: peças de teatro, partituras musicais, receitas culinárias, procedimentos organizacionais, roteiros cinematográficos, métodos algébricos, e muitos outros. Por outro lado, na nossa sociedade largamente computadorizada, convivemos diariamente com as ações executadas por sistemas de informação automatizados, e em muitos casos, somos totalmente dependentes deles. Assim, além de passarmos boa parte do nosso tempo executando algoritmos, estamos freqüentemente atendendo às determinações dos algoritmos executados por computadores, que orientam boa parte das nossas ações corriqueiras. Ou seja, estamos sempre envolvidos com *processos resultantes de um certo tipo de conhecimento acumulado e sistematizado por outrem*, e que termina por se exprimir através de algoritmos. Vistos por este ângulo, os algoritmos são expressões de um *saber fazer* (know-how), que se *descola* da realidade, e se projeta no tempo e no espaço, por meio de uma linguagem estruturada⁹⁹.

O uso de algoritmos antecede em muito a existência dos computadores, e mesmo hoje em dia, os algoritmos têm vida independente das máquinas que os executam. De fato, há muitas situações em que os algoritmos são cruciais, embora os computadores estejam ausentes: um exemplo interessante são os procedimentos de socorro a vítimas de parada cardíaca, atualmente realizados com base em algoritmos detalhados¹⁰⁰.

Todo algoritmo é composto por um conjunto ordenado de instruções, mais um conjunto de dados sobre os quais estas instruções atuam. Para elaborar um algoritmo, portanto, devemos definir os dados a serem tratados. Uma receita de bolo, por exemplo, pode falar em “250g de açúcar refinado”; pode até mesmo discriminar a marca do açúcar, mas dificilmente fará referência à safra da cana, ou à usina onde o açúcar deve ter sido produzido. Estes dois últimos elementos, a usina e a safra, são supostamente irrelevantes para os propósitos da receita e, portanto, não devem ser incluídos como

⁹⁹ Não necessariamente uma linguagem formal. A exigência pela utilização de uma linguagem formal surge quando traduzimos algoritmos em programas de computador, ou quando lançamos mão de métodos algébricos, ainda de efetuando os cálculos manualmente.

¹⁰⁰ Fartos exemplos desse uso dos algoritmos podem ser encontrados em Grauer e Cavallaro (1997).

dados do algoritmo. Para as crianças que comerão o bolo de aniversário, o que importa, em relação ao fator “açúcar”, é que o bolo esteja suficientemente doce. Por outro lado, se estivermos diante de instruções para a preparação de um requintado almoço à francesa, uma série de detalhes, antes irrelevantes, passam a ter importância fundamental. Por exemplo, a safra e a temperatura do vinho, a época em que os ingredientes foram colhidos, o posicionamento exato da louça e talheres, são elementos que compõem a atmosfera da refeição e, conseqüentemente, passam a figurar como dados essenciais. Em resumo: os itens que definiremos como dados do algoritmo dependem dos nossos propósitos. Nesse sentido, abstrair significa distinguir os *elementos relevantes* para a solução de um problema ou execução de uma tarefa, desconsiderando os demais elementos.

Ao abstrair, representamos o know-how de maneira sintética. Mais do que isso, possibilitamos que as ações descritas sejam realizadas por terceiros, independentemente da reprodução exata das condições em que elas foram originalmente executadas. Implicitamente, a definição de um algoritmo estabelece limites para as circunstâncias em que a tarefa será executada. Supõe-se que, respeitados esses limites, os resultados obtidos serão satisfatórios, mas isso às vezes não acontece.

Há alguns anos, comentou-se que uma empresa de bebidas teria decidido produzir, no Japão, uma réplica do famoso whisky "puro malte" escocês. Para tanto, teria contratado grandes especialistas do ramo, com a missão de duplicar o processo de fabricação utilizado nas destilarias da Escócia. Imaginando como o processo foi levado a cabo, suponhamos que os especialistas preparam um relatório detalhado, cujo núcleo é um algoritmo para a produção de whisky, descrevendo em minúcias todos os ingredientes, condições e procedimentos necessários para fabricar o legítimo “puro malte escocês”. Depois, providenciaram a replicação de uma destilaria típica da Escócia, em território japonês, bem como a importação de todos os ingredientes originais. Iniciada a produção, entretanto, verificou-se que o whisky resultante não tinha as características de paladar e aroma de um puro malte. Contudo, se o *algoritmo de produção* foi rigorosamente seguido, e todos os *dados relevantes* foram considerados cuidadosamente, o que pode ter saído errado?

De fato, nem todos os dados relevantes foram considerados. Conforme se comenta, análises posteriores teriam indicado que um certo caráter *indefinível* do escocês legítimo é dado pela água utilizada na sua produção. Esta água, ao descer pelas corredeiras, em regiões inabitadas da Escócia, absorveria um certo aroma característico de uma espécie de musgo ali presente. Com isso, o caráter inconfundível do puro malte escocês seria

fruto, em grande medida, de um capricho da natureza, inimitável em um outro contexto, que não aquele onde o whisky é originalmente produzido¹⁰¹.

Esse exemplo – seja real ou fantasioso – torna patente uma limitação do pensamento algorítmico, que é a dificuldade em se estabelecer, *a priori*, os elementos a serem considerados ou descartados no processo de abstração. E ainda quando tais elementos são estabelecidos com grande exatidão, eles podem se revelar impossíveis de serem transportados entre diferentes contextos. Quando elaboramos um algoritmo, portanto, corremos o risco de nos afastar dos resultados que desejaríamos obter, porque o processo de abstração está sempre sujeito a omissões irremediáveis.

Deve-se ressaltar que os algoritmos executados por seres humanos sempre deixam uma margem para que a individualidade do executor se manifeste, de modo que um algoritmo nunca será executado duas vezes da mesma maneira: um mesmo músico nunca executa uma partitura, duas vezes, de modo idêntico. Por outro lado, o senso comum faz com que tenhamos uma certa autonomia ao executar algoritmos. Por isso, não precisamos explicitar a uma pessoa, nos mínimos detalhes, tudo o que lhe pedimos para fazer. Quando, por exemplo, ensinamos a um amigo como chegar à nossa casa, de carro, damos apenas as indicações básicas, confiando em seu *bom senso* para cuidar dos detalhes, tais como engarrafamentos, buracos na pista, local de estacionamento, etc. Nós não ensinamos ao nosso amigo como virar o volante, a cada curva do caminho. Ao contrário, *abstrairmos* esse aspecto do percurso, irrelevante para os nossos propósitos. Da forma semelhante, ao escrever uma peça teatral, o autor não se preocupa em determinar a marcação exata de cada personagem. Se o fizesse, interferiria negativamente na direção do espetáculo, além de se prender a um determinado tipo de montagem. Nesse caso, o que está em jogo é a garantia de que, apesar de seguirem o roteiro (um algoritmo), diretores, atores e coreógrafos mantêm o seu papel criativo.

Contraopondo-se a essa flexibilidade, certas áreas do mundo empresarial contemporâneo transportam para o campo dos serviços as características da linha de montagem, onde se busca a máxima produtividade com o mínimo risco, através da cisão radical entre pensamento e ação: uma vez que os planejadores elaboram os algoritmos, só resta aos executores segui-los à risca, para alcançar resultados rigorosamente dentro do padrão. Em casos como o do McDonald's, o consumidor exige *o padrão* – nada mais, nada menos. Ingredientes-padrão, preparo-padrão, gosto-padrão. Mas, ao utilizar um bom

¹⁰¹ Infelizmente, não pude recuperar a fonte dessa pequena história (talvez fantasiosa), mas ainda assim optei por incluí-la, tendo em vista sua representatividade, no que tange às limitações dos algoritmos como forma de sistematizar o conhecimento.

hotel, esperamos que os atendentes e a gerência sejam capazes de lidar com eventuais alterações de última hora em nossos planos de viagem, ou com as particularidades dos nossos hábitos alimentares. Nesse caso, embora existam algoritmos definidos para as operações rotineiras do hotel, é essencial que os empregados tenham também a capacidade, a boa vontade e a liberdade para atender a demandas particulares dos hóspedes, a fim de tornar mais agradável sua estadia. Enquanto os hotéis de luxo cobram caro para nos dar a possibilidade de fugir dos algoritmos, o McDonald's e seus assemelhados cobram caro pela garantia de seguir os algoritmos à risca.

Ao discutir a tão temida substituição do ser humano pela máquina, em caráter generalizado, Collins (1992) afirma que

As máquinas não podem substituir as competências ordinárias. [...] Se nós imitarmos as máquinas em nossos atos, então as máquinas poderão nos substituir no nosso trabalho, mas este já deverá ser do tipo maquinal, para que as máquinas possam ser colocadas no lugar. Aqueles que querem substituir os homens por máquinas, no trabalho, devem antes redefinir o trabalho de tal forma que ele possa ser executado de forma maquinal; e aí que entra o processo de desqualificação¹⁰² (p.289).

Ora, as máquinas em funcionamento são algoritmos em execução. Logo, para que os algoritmos se estabeleçam como força dominante, substituindo mulheres e homens no trabalho, é necessário um esforço prévio de “desqualificação” do profissional humano. Para tanto, concorre a apologia do computador, como modelo de eficácia no mundo produtivo, que não é apenas ingênua, mas também nociva, porque desestimula os trabalhadores a pensar criticamente e criativamente.

Os algoritmos têm espaço garantido no mundo contemporâneo, como forma de comunicar conhecimentos, no campo artístico, científico, tecnológico e profissional. Isso não implica, contudo, que se deva estabelecer uma cultura centrada nos algoritmos, ou – pior ainda – um culto aos computadores, que lhes servem de suporte. Reconhecer nos algoritmos um importante auxílio para o desenvolvimento no saber, e nos computadores, o seu veículo por excelência, significa equipar as pessoas para resolver

¹⁰² Les machines ne peuvent pas remplacer les compétences ordinaires. [...] Si nos imitons les machines dans nos actes, alors les machines pourront nous remplacer dans notre travail, mais celui-ci doit déjà être du type mécanique pour que les machines puissent être mises en place. Ceux qui veulent remplacer les hommes au travail par des machines doivent d'abord rédefinir le travail de telle sorte que celui-ci puisse être effectué de façon mécanique; c'est là qu'intervient le processus de déqualification

certos problemas de forma eficaz. Mas, enxergar nos computadores e algoritmos um modelo para a organização do pensamento e do trabalho, significa tolher os passos da razão. Para que os algoritmos cumpram o seu papel legítimo, cabe às pessoas definir de que forma eles tomarão parte na vida social: esta é só uma, dentre as incontáveis tarefas que os computadores não podem fazer por nós.

5.2. Implicações pedagógicas

A disseminação dos computadores, rodando software cada vez mais complexo, suscita vários desafios para a educação. Estão em jogo, entre outras questões, o nosso posicionamento crítico diante dos computadores, bem como a capacidade de fazer bom uso dos resultados obtidos através deles.

Collins (1992, p.93) narra um caso simples de postura acrítica diante de uma máquina de calcular, representativo do que poderia ser uma postura acrítica diante dos computadores, em situações mais delicadas. Trata-se de um aluno que precisava calcular a posição de um ponto em uma reta, bastando, para isso, multiplicar a coordenada x , igual a 352, pela inclinação da reta, igual a $-0,84104$. Seguindo seu “bom senso”, o aluno, “pouco versado em números”, digitou “ $352 \times -0,84104 =$ ”, e aceitou prontamente a resposta: 351,15896. Obviamente uma resposta errada, pois a resposta certa seria $-296,04608$. Porém, da forma como a operação foi inserida na calculadora, o que o aluno realmente fez foi subtrair 0,84104 de 352, porque o sinal de subtração se sobrepôs ao de multiplicação, digitado antes: daí a resposta incorreta.

Partindo desse exemplo, algo extremo, podemos pensar em quantos engenheiros inexperientes estariam aptos a criticar os resultados fornecidos pelos programas de cálculo estrutural. Ou ainda, quantos químicos poderiam detectar falhas sutis na análise topológica de moléculas, realizada por software especializado. Ou, quantos enxadristas teriam condições de detectar os erros de análise de um programa de xadrez muito forte – e assim sucessivamente. O ponto crucial é este: na medida em que os computadores substituem os seres humanos, em atividades que envolvem conhecimento, reduz-se o número de pessoas qualificadas para criticar os resultados obtidos pelos programas.

Ao compreender em que princípios se baseiam os programas de xadrez, um enxadrista estará apto a entender porque eles jogam de determinada maneira. Não que o jogador, ao compreender os programas, consiga derrotá-los, necessariamente. Ao contrário, é sabido que os melhores programas de xadrez têm superado, por larga margem, a

destreza enxadrística dos seus programadores (Levy e Newborn, 1991, p.223). Ainda assim, se um jogador mediano entende os programas, ele entende *porque perde*, e com isso se elimina o risco de interpretações ingênuas para o desempenho da máquina, como acreditar que ele se deve a uma performance inteligente.

O jogador que entende os programas sabe que tipo de falhas eles podem cometer, e que tipos de falhas provavelmente nunca cometerão. Na prática, isso significa, por exemplo, que um jogador de xadrez postal não confiará cegamente na avaliação de um programa para uma posição estrategicamente complicada, sem variantes forçadas, e que requer a elaboração de um plano de longo prazo. Ele também observará, na tela de análises do programa, o nível de profundidade da árvore de busca, a fim de identificar situações críticas que eventualmente poderiam escapar ao horizonte de cálculo da máquina.

O conhecimento a respeito dos algoritmos e computadores cresce em importância, na medida em que as interfaces homem-máquina simulam comportamentos inteligentes, dissimulando as limitações das máquinas. Alguns dos melhores programas de xadrez possuem, hoje em dia, um módulo de análise capaz de anexar comentários por escrito aos jogos. É uma função útil para treinamento, pois os comentários refletem aspectos de análise posicional de que as heurísticas dão conta, mas que muitas vezes passam despercebidos até mesmo para jogadores humanos de bom nível, no contexto de uma determinada partida. Além disso, as variantes inseridas juntamente com os comentários são bastante precisas, porque decorrem do poder de cálculo do computador. Porém, deve ficar claro que as análises da máquina não substituem, de forma alguma, o julgamento de um enxadrista de alto nível.

A avaliação da máquina, por mais que pareça inteligente, limita-se aos padrões que a função de avaliação heurística é capaz de reconhecer, dentro do horizonte de cálculo permitido pelo hardware. A própria linguagem em que os comentários são redigidos é estereotipada, porque as frases são compostas a partir de umas poucas estruturas sintáticas previamente definidas. A máquina também não pode dialogar sobre suas avaliações, nem apresentá-las sob diferentes perspectivas, como fazem os bons comentaristas. A avaliação da máquina para uma posição de xadrez é quantitativa, e se exprime através de números, ao passo que o olhar de um grande mestre é qualitativo, e se exprime por meio de conceitos culturalmente contextualizados. Assim, ao passo em que a máquina faz comentários genéricos sobre o posicionamento das peças, domínio do centro, etc., um grande mestre aponta planos, sugere caminhos alternativos, e destaca os aspectos específicos da posição, relevantes para o desenvolvimento da partida.

Podemos considerar que, ao jogar xadrez, um grande enxadrista assemelha-se em parte a um compositor erudito, e em parte a um arquiteto, que concretizam suas obras a partir de núcleos temáticos bem definidos, com o suporte da técnica, nas suas respectivas áreas de atuação. Ao se igualar à máquina, enveredando pelo cálculo indiscriminado de variantes, e pela busca de padrões de jogo pré-definidos, o enxadrista se assemelha a um arquiteto sem visão de conjunto, ou um compositor carente de tema; em suma, é um artífice que perdeu o domínio da obra. Infelizmente, muitos livros de xadrez recentes pautam-se nesse tipo de abordagem *algoritmizada*, tentando conduzir o leitor por um labirinto de variantes e heurísticas (no fundo, *macetes* táticos e posicionais), de forma a possibilitar a obtenção de resultados competitivos em curto prazo.

Os livros clássicos, antigos e recentes, são diferentes. Enfatizam os comentários profundos e destacam as sutilezas do jogo, estimulando o estudante a buscar seus próprios caminhos. Nessa linha, um dos maiores modelos é o livro de David Bronstein (1984) sobre o Torneio de Zurique 1953. Nessa obra, o cálculo de variantes se resume ao indispensável, inserindo-se no bojo de comentários verbais elegantes e inspiradores. A título de ilustração, acompanhemos dois comentários, escolhidos, devido à sua relativa simplicidade, em meio ao farto material de qualidade que com que Bronstein nos brinda.

Primeiro, a posição ilustrada no Diagrama 4, que ocorreu na partida Szabo x Petrosian¹⁰³, na nona rodada do referido torneio. O último lance das Brancas foi **19.b3**, a respeito do qual Bronstein comenta (p.163):

A idéia fatal de atacar o ponto **d5** desde **c4** passa pela cabeça de Szabo e com certa velocidade ele faz uma jogada que enfraquece todo um conjunto de casas negras em seu campo de jogo, minando o suporte do cavalo **c3** e ativando o bispo negro, que obtém, assim, possibilidades de se deslocar a **b4** ou **a3**.¹⁰⁴

¹⁰³ No cabeçalho das partidas de xadrez, o nome do condutor das peças brancas sempre aparece à frente do nome do condutor das peças negras.

¹⁰⁴ La fatal idea de atacar **d5** desde **c4** cruza la mente de Szabo y con cierta ligereza efectúa una jugada que debilita todo um grupo de casillas negras em su campo, minando el soporte del caballo **c3** y activando el alfil negro, qui obtiene así chances de acceder a **b4** o **a3**.

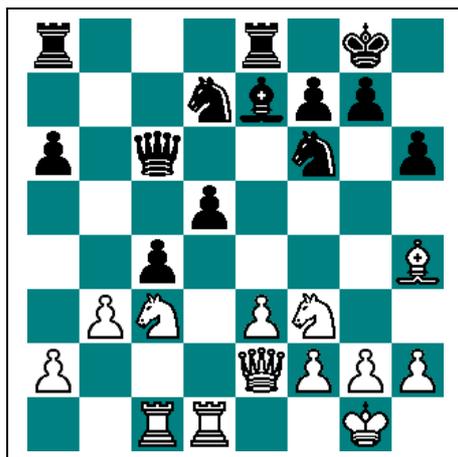


Diagrama 4: Zurique 1953 - Szabo x Petrosian

Reparemos que Bronstein não diz nada sobre “padrões”, e nem discute heurísticas. Ao invés disso, faz um comentário sucinto e bem delimitado, específico para posição em foco, mas que ao mesmo tempo aguça o senso posicional do leitor, evocando uma visão de conjunto que será útil em muitas outras circunstâncias. Com isso, Bronstein desafia o leitor a apreender um dos principais caracteres distintivos de um grande jogador, que é a habilidade de capturar o caráter essencial, os elementos posicionais relevantes de uma posição, antes de se envolver com o cálculo de variantes.

A seguir, o comentário que se segue ao lance 23 das Brancas, na partida Reshevsky x Averbach, na 18ª rodada do Torneio:

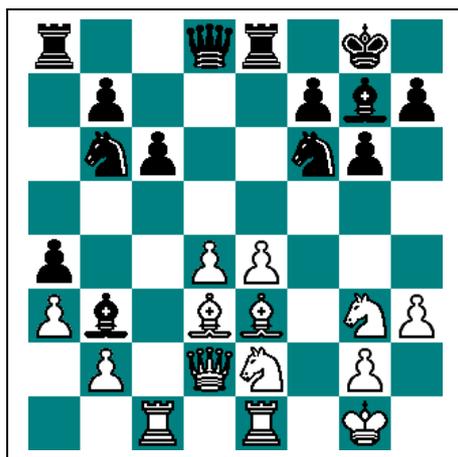


Diagrama 5: Zurique 1953 - Reshevsky x Averbach

As Brancas construíram um forte centro. A próxima fase do plano é atacar o rei, mas ainda não chegou a hora das peças entrarem na batalha: o ataque há

de começar com o avanço do peão **h** para abrir a muralha da fortaleza¹⁰⁵ (p.333).

Aqui, novamente, vê-se o desenvolvimento lógico de uma idéia, aplicada a uma posição específica. Vale notar que Bronstein afirma que ainda não chegou a hora do ataque com as peças, mas não desce a minúcias para justificar a afirmativa. É uma percepção sua, que cabe ao leitor investigar mais a fundo, novamente sem o apoio de heurísticas e “padrões” predeterminados.

São muitos os livros que, tal como o de Bronstein, dispensam a abordagem algoritmizada do xadrez, priorizando a explanação dos fundamentos do jogo, a partir da análise meticulosa de um certo número de exemplos. Dentre outros clássicos, podem ser citados “The Middle Game in Chess” (Znosko-Borovsky, 1930), o “Tratado General de Ajedrez” (Grau, 1958; 1959) e, mais recentemente, “Endgame Strategy” (Shereshevsky, 1985). Tal como o livro de Bronstein, estes estão cheios de observações perspicazes e idéias inspiradoras, e é por isso que conservam o seu valor, durante décadas a fio.

O conhecimento enxadrístico, tal como o conhecimento científico, progride a partir de hipóteses e refutações, idéias criativas e testes sistemáticos. Tal como o conhecimento científico, o conhecimento enxadrístico está sujeito aos paradigmas dominantes de cada tempo. Frequentemente, certas linhas de jogo, tidas como irremediavelmente refutadas, voltam à cena fortalecidas, com excelentes resultados¹⁰⁶. Em contrapartida, linhas que gozam de grande popularidade durante algum tempo são depois descartadas, em função de sucessivos resultados desfavoráveis, em competições de alto nível¹⁰⁷.

Partidas de xadrez clássicas, às vezes com mais de um século, continuam sendo estudadas por jogadores de todos os níveis, porque se desenvolvem em torno de eixos temáticos que não perdem a atualidade. Elas podem ter sido superadas, em termos de lances específicos, após exaustivas análises e experiências, no decorrer do tempo. Contudo, o impulso criativo que as gerou é perene. Por isso, os estudiosos continuam a

¹⁰⁵ Las blancas han construído un fuerte ciento. La próxima fase del plan és atacar al rey, pero aún no ha llegado la hora de que las piezas entren en la batalla: el ataque ha de comenzar con el avance del peon **h** para abrir la muralla de la fortaleza.

¹⁰⁶ É o caso da Variante Berlinense, da abertura Ruy Lopez, que durante décadas quase não foi praticada. No seu match com Kasparov, Victor Kramnik ressucitou esta variante, mantendo a paridade em todas as partidas, com as peças negras.

¹⁰⁷ A exemplo da Defesa Índia do Rei, que até a década de 1980 era uma das aberturas mais jogadas por grandes mestres de primeira linha, mas que, atualmente, quase não aparece nos principais torneiros.

aprender com as idéias que motivaram os lances, mesmo que alguns deles, com o passar dos anos, tenham se demonstrado insustentáveis, na prática competitiva. Ao analisar as partidas clássicas, os computadores eventualmente apontarão esse ou aquele lance ruim, em termos posicionais, assim como os erros táticos – mas sempre de forma contingente. Só um jogador experiente, com elevado nível de *cultura enxadrística*, pode seguir o *fio condutor* dos sucessos e revezes de seus antecessores. É por isso que nenhum autômato substituiu os livros clássicos, nem tampouco os bons professores.

As máquinas são importantes para checar a precisão das análises humanas, mas só depois que os planos de jogo para uma determinada posição tiverem sido equacionados. Para os jogadores inexperientes, o hábito de analisar partidas com a assistência dos programas pode ser bastante prejudicial, pois dirige o raciocínio para o cálculo frenético de variantes, muitas das quais desprovidas de sentido estratégico. Com suas enormes bibliotecas de partidas, prontamente acessíveis, os computadores também facilitam a memorização de aberturas em estado bruto, isto é, sem o estudo paralelo das idéias que as fundamentam. Então, ao acumular o máximo de conhecimento enciclopédico, para alcançar resultados imediatos nas competições, os jovens enxadristas obstruem seu próprio aprendizado de longo prazo¹⁰⁸. Conseqüentemente,

É preferível se possuir uma pequena soma de conhecimentos, flexível, entendida e sistematizada, do que uma enorme massa de fatores isolados, que representam um peculiar capital morto enxadrístico. Ler-se as entradas de um dicionário, uma após a outra, não é a melhor forma de se desenvolver o intelecto¹⁰⁹ (Krogius, 1974, p.46).

Donde se conclui que o aprendizado do xadrez não é possível apenas com o apoio das máquinas, e menos ainda, tomando-as como modelo. Ao contrário, o aprendizado no xadrez, assim como na ciência e na tecnologia, é um processo que só se integraliza em um contexto cultural, que envolve dois aspectos críticos: (1) uma coletividade, ou *comunidade*, que compartilha certos interesses e problemas, e (2) idéias e *visões*, que se

¹⁰⁸ Há muitos jogadores de rating elevado que, devido às pressões profissionais, caem vítimas dessa abordagem pragmática, que associa heurística e memorização. Não raro, demonstram surpreendentes lacunas conceituais, resultando em atuações decepcionantes em relação ao seu suposto nível de jogo – à semelhança do que acontece com os computadores.

¹⁰⁹ Es preferible poseer una pequeña suma de conocimientos, flexible, entendida e sistematizada a una enorme masa de factores aislados que representan un singular capital muerto ajedrezista. Pues leerse uno tras otro los artículos de un diccionario enciclopédico no es el mejor método para desarrollar el intelecto

materializam sob a forma de hipóteses e planos de trabalho, métodos e técnicas, concorrendo para a consecução de certos fins¹¹⁰.

Todos os grandes jogadores da história recente do xadrez (da década de 1940 em diante), com raríssimas exceções, foram oriundos de grandes escolas, ou, ao menos, tiveram o acompanhamento próximo de tutores altamente qualificados¹¹¹. Esse fato evidencia que o estudo enciclopédico, apenas, não basta para desenvolver um xadrez de alto padrão; ao contrário, a imersão em um ambiente de cultura enxadrística bem desenvolvida é fundamental. Jogadores como o norte-americano Bobby Fischer¹¹² e o Brasileiro Henrique Mecking¹¹³ (Mequinho) não chamaram a atenção do mundo apenas por sua precocidade, mas porque se formaram fora das grandes escolas¹¹⁴.

¹¹⁰ A noção de interação social, como condição necessária para o aprendizado em ciência e tecnologia, é amplamente desenvolvida por Collins (1992) e Collins e Kusch (1998). Roszak (1988) enfatiza a prevalência das idéias sobre os dados e informações, na constituição de uma cultura. Embora originalmente formulada tendo em mente um contexto social amplo, a tese de Roszak se aplica à reflexão sobre as comunidades científica e tecnológica, assim como a comunidade enxadrística.

¹¹¹ Quando falamos em “escolas” de xadrez, não nos referimos às academias para o ensino do jogo, mas a uma tradição cultural, onde as academias estão inseridas. Assim, podemos falar na “escola soviética de xadrez”, na “escola britânica”, e assim por diante. A importância da tradição é tão grande, que até hoje a esmagadora maioria dos melhores jogadores do mundo está vinculada à Rússia e aos demais países da antiga “Cortina de Ferro”. O Prof. João Carlos Salles, em uma conversa informal, lembrou que mesmo com todos os atuais recursos para a difusão da cultura enxadrística – incluindo os programas e a Internet – os jogadores que emigram do leste europeu tendem a perder algo da sua capacidade competitiva. Dos vinte jogadores mais bem situados no ranking da FIDE (2002), nove são russos, dois ucranianos, dois húngaros, um armênio, um búlgaro, um indiano, dois israelenses, um espanhol e um inglês, sendo que o espanhol e os israelenses são, na verdade, imigrantes de origem soviética. Ou seja, os jogadores ligados às escolas do Leste Europeu ainda dominam, de forma absoluta, a cena no xadrez mundial.

¹¹² Robert James Fischer, nascido em 1943, é considerado um dos maiores gênios criativos da história do xadrez. De 1948 até o presente, foi o único jogador não “soviético” a se sagrar campeão do mundo, título que manteve entre 1972 e 1975.

¹¹³ Henrique da Costa Mecking, nascido no Brasil, em 1952, foi criança-prodígio, e por duas vezes chegou às quartas de final na disputa pelo título mundial de xadrez. Durante um breve período, no início dos anos 70, Mecking, então com aproximadamente vinte anos, permaneceu entre os cinco enxadristas mais bem situados no ranking mundial.

¹¹⁴ O Clube de Xadrez de Manhattan, onde o talento de Fischer se desenvolveu, tinha alguma tradição, mas em nada comparável às academias de xadrez do Leste Europeu. Tanto Fischer quanto Mequinho têm em comum o fato de, a partir dos quinze anos de idade, nunca terem tido adversários à altura, nos seus países de origem. Isso não aconteceu com nenhum jogador dos países com forte tradição enxadrística, em

Sob uma perspectiva de ensino-aprendizagem, a forma como os computadores jogam xadrez, pautada em regras e cálculos, embora eficaz, é irrelevante como modelo para os seres humanos¹¹⁵. A idéia de que o desempenho das pessoas, ao resolver problemas, se pauta em seguir certas regras, ainda que de modo inconsciente, não se sustenta, mesmo em um domínio formal, como o xadrez. Esta questão relaciona-se com a diferenciação entre “saber o que”, e “saber como”. Assim,

Faz sentido perguntar em que momento alguém se apercebeu de uma verdade, mas não perguntar em que momento alguém adquiriu uma habilidade. “Parcialmente-treinado” é uma expressão com significado, mas “parcialmente-informado” não é¹¹⁶ (Ryle, 1984, p.59).

Ou seja, devemos dizer sempre que determinada pessoa tem ou não tem conhecimento de uma certa regra, pois não faz sentido dizer que a pessoa sabe uma regra pela metade. Contudo, podemos dizer que alguém possui essa ou aquela habilidade em maior ou menor grau, porque a aquisição de habilidades é um processo gradativo.

Um instrutor de auto escola pode enunciar para o seu aluno uma série de regras importantes para a condução de veículos, que serão aprendidas, mas nem por isso o aluno vai necessariamente dirigir bem. O instrutor pode dizer, por exemplo, que não se deve frear abruptamente em chão molhado. Ainda assim, o aluno eventualmente cometerá esse erro se, por imperícia, não observar o estado do pista, ou se assustar com algum imprevisto no trânsito, ou ainda, se não tiver a sensibilidade para dosar a pressão do pé sobre o pedal do freio. Então, embora o aluno conheça a regra, ainda não tem a habilidade para agir convenientemente, ao dirigir um carro.

Assim podemos, eventualmente, combinar afirmativas aparentemente contraditórias, como: (1) O enxadrista João *sabe* que, ao conquistar uma vantagem de espaço, com peças mais ativas do que as do adversário, deve atacar, e (2) O enxadrista João freqüentemente não ataca no tempo certo, após conquistar uma vantagem de espaço, com peças ativas. Na verdade não há aqui contradição alguma, porque, na afirmativa (1)

especial a antiga União Soviética, Grã-Bretanha, Hungria e Iugoslávia.

¹¹⁵ O que não significa que as máquinas de xadrez sejam irrelevantes. Na verdade, elas são da maior importância para o desenvolvimento das habilidades enxadrísticas dos seres humanos, conforme discutido na seção 5.3 da presente dissertação.

¹¹⁶ It makes sense to ask at what moment someone became apprised of a truth, but not to ask at what moment someone acquired a skill. ‘Part-trained is a significant phrase, ‘part-informed’ is not.

estamos simplesmente dizendo que o enxadrista em questão *está ciente de uma regra*, e na afirmativa (2), fica implícito o fato de que ele não tem a *habilidade prática* necessária para se portar de acordo com a regra, em todas as situações concretas, perante o tabuleiro. Tal como a maioria dos jogadores de força mediana, o enxadrista João provavelmente conhece muitas heurísticas (está informado, “sabe o que”), mas tem dificuldades em aplicá-las de forma contextualizada (não é suficientemente hábil, apenas “parcialmente-treinado”, ou seja, “sabe como” apenas parcialmente). Conhecer regras¹¹⁷, portanto, não basta para jogar bem o xadrez, mas não apenas porque “todas as regras têm exceções”. O problema é que não se pode conceber, indefinidamente, regras sobre regras, em sucessivos níveis de detalhe, de modo a contemplar satisfatoriamente todas as situações com que o enxadrista pode se defrontar.

Na medicina, os cirurgiões experientes despertam maior confiança do que os cirurgiões recém-saídos da residência médica. Entretanto, a teoria a que ambos têm acesso, na literatura especializada, é basicamente a mesma, e a habilidade manual e o equilíbrio emocional dos jovens não são necessariamente inferiores. Por outro lado, o cirurgião mais experiente é, em geral, capaz de trabalhar de modo mais espontâneo, rápido e eficaz. Superado o período de aprendizado da sua arte, ele já não precisa se lembrar das heurísticas, e executa naturalmente, os procedimentos pertinentes, caso a caso. Metaforicamente, o que diferencia o cirurgião inexperiente do cirurgião experiente é o fato de que o primeiro se apóia no conhecimento adquirido para se movimentar, com menores riscos, em um labirinto algo misterioso, enquanto o segundo caminha por trilhas familiares, usando o conhecimento teórico como apoio eventual.

No xadrez há uma história antiga que ilustra, de forma divertida, a habilidade extraordinária dos jogadores de elite, para perceber as nuances sutis do jogo, e elaborar soluções em circunstâncias onde as heurísticas não se aplicam claramente: em meio a um torneio internacional de alto nível, no início do século XX, alguns jogadores com nível de mestre estavam discutindo acaloradamente uma determinada posição. O problema consistia em saber se, no lance seguinte, o jogador das peças Brancas deveria ou não entregar o seu bispo em troca da torre adversária. Sendo a torre uma peça de valor superior ao bispo, e não havendo ameaças táticas visíveis, impedindo a troca, apenas fatores posicionais mais profundos poderiam eventualmente alterar a decisão. A dúvida era, justamente, se valia à pena para o jogador das Brancas optar pelo ganho material, ainda que dando às peças negras uma certa vantagem posicional. Enquanto a discussão prosseguia, sem solução à vista, passou por perto Emmanuel Lasker, o

¹¹⁷ Refiro-me a regras que compõem heurísticas, e não às regras formais do jogo (regulamento).

campeão do mundo, à época¹¹⁸. Respeitosamente, e algo ansioso, um dos mestres expôs rapidamente o problema a Lasker, concluindo com a questão: “o bispo deve ou não deve tomar a torre?” Em resposta, Lasker olhou rapidamente para o tabuleiro e, com um gesto de enfado, respondeu: “uma torre dessas não se toma”. E saiu, deixando os mestres, agora mais perplexos do que antes¹¹⁹. Ora, enquanto os mestres se perdiam em argumentos contra ou a favor da tomada da torre, Lasker captou o sentido da posição que lhe foi apresentada, dando um parecer para ele tão óbvio que não carecia de justificação¹²⁰.

Certa feita, um amigo enxadrista, várias vezes campeão baiano, me mostrava uma partida de xadrez que ele achara muito interessante. Em determinado momento, ele me disse: “veja como o jogador das Negras protege a quarta linha contra a penetração das peças adversárias”. Ao que eu respondi, “muito interessante, onde foi que você viu esse comentário?” E ele me respondeu, “em lugar nenhum, eu mesmo observei”. Ora, meu amigo inventou *uma forma de ver a posição*, em meio a outras tantas formas possíveis, o que vai muito além do que um programa baseado em heurísticas pode fazer.

Jogar bem o xadrez requer o desenvolvimento de certas faculdades intelectuais de base, especialmente a percepção, o juízo e a inventividade. Requer ainda a apreensão do que é específico em cada partida, mas ao mesmo tempo, exige a aplicação de leis gerais em situações concretas. Para um enxadrista qualificado, jogar significa exprimir todo um universo cultural¹²¹ através da sintaxe restrita dos lances, sobre o tabuleiro. Como o

¹¹⁸ Emanuel Lasker (1868-1941), nascido na Prússia, é considerado um dos maiores jogadores de todos os tempos, tendo se mantido como campeão mundial durante 27 anos, de 1894 a 1921.

¹¹⁹ Esta história é bem conhecida na comunidade enxadrística, sendo narrada em mais de um livro. Infelizmente, não pude localizar uma fonte específica da narrativa, que, contudo, expressa de modo característico a diferença entre jogadores de elite e jogadores de segunda linha.

¹²⁰ Analogamente, ao longo de uma demorada e difícil cirurgia, o cirurgião-chefe não tem que justificar para si mesmo cada um dos seus procedimentos. Mas, se a cirurgia for gravada em vídeo, para fins didáticos, ele procurará explicar os seus procedimentos por meio de regras. Contudo, nem o mais experiente dos cirurgiões e nem o mais experiente dos enxadristas podem ter certeza de todos os seus passos, e portanto, não podem definir regras absolutas para justificar suas ações. Na engenharia do conhecimento, é sempre difícil se extrair, dos grandes especialistas, regras que justifiquem a sua conduta.

¹²¹ Há o universo cultural próprio do xadrez, com sua história, seus conceitos, sua terminologia e seus problemas. Mas há também a interpenetração do xadrez com outros aspectos da cultura. Em entrevista recente, o veterano grande mestre russo Mark Taimanov (2002), ex-candidato ao título mundial e grande pianista erudito, estabelece um nexos entre o estilo dos seus enxadristas preferidos e seu gosto pelos

xadrez se pratica em grupos, a capacidade de dialogar sobre idéias e convicções é também fundamental. Tudo isso, uma pessoa que cultiva o xadrez vai aprendendo, mesmo que não jogue muito bem. Mas, nada disso um computador faz ou ensina a fazer, por melhor que seja o seu desempenho nos jogos de competição.

Não se deve antropomorfizar os computadores, nem tampouco mecanizar o homem. As heurísticas e os cálculos ultra-rápidos são uma tentativa de colocar pensamento na máquina. Até hoje, porém, o pensamento não “coube” na máquina. Na via oposta, não há porque inverter o processo criativo dos programadores, tentando limitar o pensamento humano ao que a máquina pode fazer. À educação, portanto, colocam-se dois desafios: O primeiro, estimular a reflexão sobre os computadores, para uma melhor compreensão de seus limites e possibilidades. O segundo, promover o desenvolvimento das faculdades de percepção e juízo, diálogo e inventividade, que nos fazem diferentes das máquinas, e que nos permitem utilizá-las com maior proveito.

5.3. Ferramenta epistemológica

Os usos mais comuns do computador não apresentam novidades, no plano epistemológico. Nas atividades burocráticas, eles executam procedimentos rotineiros, sem características de investigação ou descoberta. Como instrumentos midiáticos, permitem que as informações circulem com uma dinâmica sem precedentes na história, cumprindo o papel de interfaces sofisticadas, agregando características de televisão, telefone, fax e telex, em um só equipamento. Funcionando como portal para a WEB, o computador transmite conhecimento, mas não participa ativamente de novas descobertas, senão como meio para a comunicação de descobertas.

O computador é uma ferramenta epistemológica na medida em que opera como instrumento ativo na produção do conhecimento. Ao processar dados com inimitável eficiência, o computador permite a aplicação dos modelos matemáticos a problemas inacessíveis por outros meios, revolucionando a investigação dos fenômenos naturais, assim como a de domínios abstratos, a exemplo do xadrez. Se a solução para um problema puder ser definida algorítmicamente, e se o algoritmo puder ser implementado de forma eficiente, então o computador ajudará a resolver o problema. Por exemplo, os métodos para resolução dos sistemas lineares de múltiplas variáveis são conhecidos há séculos. Contudo, eles requerem a repetição de operações aritméticas, em quantidade proporcional ao número de variáveis do sistema, inviabilizando a solução manual para

compositores românticos.

sistemas com umas poucas dezenas de variáveis. Com o auxílio do computador, sistemas com milhares de incógnitas podem ser resolvidos, o que implica em soluções inéditas para problemas cruciais em várias áreas da ciência e da tecnologia.

O potencial do computador como ferramenta epistemológica fica evidente nas simulações. Em muitos casos, somos capazes de descrever satisfatoriamente, por meios analíticos, o comportamento dos fenômenos que queremos investigar, mas freqüentemente isso não é possível. Este problema foi colocado por Averill e Kelton (1991), nos seguintes termos:

Se as relações que compõem o modelo forem suficientemente simples, pode ser possível se utilizar métodos matemáticos (tais como a álgebra, o cálculo ou a teoria da probabilidade), a fim de se obter informações exatas sobre questões de interesse; isso se chama solução analítica. Contudo, a maioria dos sistemas do mundo real são complexos demais para serem analisados analiticamente, e tais modelos têm que ser estudados por meio de simulações. Em uma simulação nós usamos o computador para avaliar um modelo numericamente, e os dados são recolhidos a fim de se estimar as características desejadas do modelo.¹²²

Donde se conclui que a simulação é uma necessidade, porque, tanto nas ciências como na matemática pura, apenas uma parcela dos problemas pode ser resolvida por meios algébricos. Na medida em que representam satisfatoriamente os fenômenos naturais, ou os espaços abstratos sob investigação, os algoritmos numéricos são portas que se abrem em certas *paredes epistemológicas*, antes intransponíveis.

Os simuladores de vôo são usados há décadas, por todas as grandes companhias aéreas. Das versões primitivas de trinta anos atrás, eles evoluíram para os sofisticadíssimos modelos atuais, que testam a fundo a habilidade dos pilotos. Os simuladores modernos simulam pousos e decolagens nos mais diversos aeroportos, sob diferentes condições de clima e movimentação no espaço aéreo. São, portanto, equipamentos de valor inestimável para seus usuários, pois representam uma enorme economia de recursos,

¹²² If the relationships that compose the model are simple enough, it may be possible to use mathematical methods (such as algebra, calculus or probability theory), to obtain exact information on questions of interest; this is called an analytic solution. However, most real-world systems are too complex to allow realistic models to be analyzed analytically, and these models must be studied by means of simulation. In a simulation we use a computer in order to evaluate a model numerically, and data are gathered in order to estimate the desired true characteristics of the model.

bem como a redução significativa dos riscos: ao pilotar uma aeronave pela primeira vez, já estão amplamente familiarizados com o seu comportamento, mesmo em condições de emergência.

As simulações também têm permitido aos engenheiros optar por soluções técnicas audaciosas, que anteriormente eram descartadas em prol de outras, mais conservadoras, a fim de diminuir riscos. Jean-Paul Teyssandier e seus colaboradores apresentam um caso notável, onde a simulação aplicada à engenharia civil viabilizou uma solução inovadora (Teyssandier, Combault e Pecker, 2000). Trata-se da construção de uma ponte sobre um braço de mar cujo leito é lamacento, em uma área sujeita a abalos sísmicos freqüentes. A fim de viabilizar economicamente o projeto, mantendo os riscos em níveis aceitáveis, os engenheiros recorreram a uma simulação inédita do comportamento dos pilares de sustentação da ponte. Várias soluções, novas e antigas, foram testadas em computador, antes de se optar pela mais viável. Deve-se enfatizar que essa análise não poderia ser feita sem o auxílio do computador, porque, por um lado, os desenhos estáticos seriam insuficientes e, por outro lado, testes práticos seriam inviáveis economicamente. Portanto, a simulação proporcionou conhecimento realmente novo, cuja qualidade, como se pode deduzir, será tanto maior quanto mais adequados forem os modelos físicos que lhe servem de suporte.

No campo propriamente científico, as simulações têm como principal função testar hipóteses de trabalho, em complemento às experiências de laboratório e observações de campo. O status da simulação nas ciências vem evoluindo rapidamente, especialmente no campo da biologia, da neurologia e da cosmologia. Estas áreas têm em comum a dificuldade para a realização de experimentos críticos que envolvem restrições éticas, ou ainda, dificuldades práticas difíceis de resolver. As simulações são usadas a fim de se superar, em parte, tais dificuldades. Como benefício adicional, elas permitem analisar, em horas, dias ou minutos, fenômenos que, às vezes, ocorrem no decurso de meses, anos, ou até bilhões de anos. Uma vez que a base para as simulações, em ciência, são leis que se supõe *fundamentais*, eventuais discrepâncias entre as simulações e os fatos observados são indícios de falhas, ou nas observações, ou na compreensão dos fenômenos. Dessa forma, mesmo quando as simulações não geram conhecimento novo, constituem-se em valioso instrumento para testar teorias.

O “Teorema das Quatro Cores”, formulado em 1852, por Francis Guthrie, afirma que qualquer mapa pode ser colorido com apenas quatro cores, de forma que nenhum par de países fronteiros apareça na mesma cor. Durante 124 anos, os matemáticos procuraram em vão provar esse teorema, até que, em 1976, Appel e Haken (1977)

apresentaram uma prova, que só foi possível graças ao uso extensivo de recursos computacionais – na realidade, mais de 1.000 horas de computação, em computadores de grande porte da época. O trabalho de Appel e Haken causou forte impacto na comunidade matemática, porque, nas palavras dos autores,

As provas tradicionais de teoremas matemáticos são razoavelmente curtas e altamente teóricas – quanto mais poderosa a teoria, mais elegante a prova – e revisá-las à mão é normalmente o melhor método¹²³ (p.121).

Portanto, o aparecimento de uma prova algorítmica para um teorema clássico, extremamente extensa e impossível de ser revisada à mão, desafia toda uma tradição intelectual. Em contrapartida, trata-se de um aporte metodológico valioso, que amplia os horizontes da mais teórica das disciplinas, de certa forma considerada um bastião do pensamento puro. A partir de 1976, paradoxalmente, a matemática pura também tem um ramo experimental.

Para um aficionado do xadrez computacional, a convergência entre pensamento e experimento, em uma área 100% abstrata, evidencia-se por si mesma. No xadrez, as idéias e conceitos estão em permanente tensão com as possibilidades concretas que cada posição oferece. O cálculo de variantes é vazio e ineficaz, quando desprovido de uma idéia que lhe sirva de linha mestra. Por outro lado, as idéias só prosperam se passíveis de realização, lance após lance, de acordo com as possibilidades que a sintaxe do jogo oferece. Calculando variantes com enorme velocidade, os programas de xadrez são um meio importante para verificar a aplicabilidade das idéias a posições específicas, de forma análoga ao que ocorre na prova de Appel e Haken. Aliás, eles reconhecem claramente a identificação entre seus métodos e os dos programadores de xadrez, ao afirmar que

Quando nós tínhamos checado manualmente as análises produzidas pelas versões mais antigas do programas, nós tínhamos sempre podido prever o seu curso, mas agora *o computador estava agindo como uma máquina de jogar xadrez*. Ele estava elaborando estratégias baseadas em todos os truques que lhe tinham sido ensinados, e as novas abordagens eram freqüentemente muito mais inteligentes do que as que tínhamos tentado¹²⁴ (p.117, grifo meu).

¹²³ Traditional proofs of mathematical theorems are reasonably short and highly theoretical—the more powerful the theory, the more elegant the proof—and reviewing them by hand is usually the best method.

¹²⁴ When we had hand-checked the analyses produced by the early versions of the program, we were always able to predict their course, but now *the computer was acting like a chess-playing machine*. It was

Malgrado o seu teor antropomórfico, esta passagem apresenta claramente o computador como uma ferramenta epistemológica, que participa ativamente na investigação de um espaço abstrato, com sintaxe própria. No xadrez, ele ajuda os jogadores a superar seus limites cognitivos, calculando exaustivamente variantes complexas, onde a capacidade analítica do ser humano se esgota, em meio a um emaranhado de possibilidades. Os computadores funcionam, portanto, como uma espécie de prótese mental, viabilizando análises cuja complexidade desafia os limites do cérebro. Com efeito, é comum que na análise de partidas de xadrez, auxiliada por computador, revelem-se detalhes que durante anos passaram despercebidos aos comentaristas.

Appel e Haken não descartam a possibilidade de que se encontre uma prova “tradicional” para o teorema das quatro cores. No xadrez, contudo, o fenômeno da explosão combinatória nos impede de pensar em “provas”, a não ser para posições de caráter muito definido. Por isso, o xadrez continuará sendo objeto de debate e experimentação, ainda por muitos anos, mesmo com o auxílio dos computadores.

A recente introdução do xadrez avançado (“advanced chess”) representa o reconhecimento da elite enxadrística do papel do computador como ferramenta para testar idéias. Nessa modalidade de jogo, cada jogador joga acompanhado de um computador, em que são instalados vários “motores” de análise, uma base de dados de aberturas e uma base de dados de finais, que podem ser usados livremente durante a partida (ChessBase, 2002). Assim, os adversários, auxiliados pelos programas, desoneram a sua memória e reduzem o tempo despendido com o cálculo mental de variantes, podendo se concentrar mais na concepção estratégica do jogo. Dessa forma, supõe-se que o nível das partidas se torne mais elevado, atingindo patamares que nem o ser humano sozinho, nem os computadores, sem a assistência humana, poderiam atingir. Para aumentar o interesse pela nova modalidade, as imagens das telas dos computadores são colocadas em telões, de forma que o público acompanha em tempo real o trabalho dos grandes mestres.

O xadrez assistido por computador não substitui o fascínio do xadrez tradicional, nem tampouco o desafio das competições entre seres humanos e computadores. Nesse contexto, devemos lembrar que o xadrez postal (ou por e-mail) continua com milhões de adeptos, e que nunca se tornou desinteressante, devido ao uso dos computadores. São modalidades distintas, cada qual com o seu papel para o desenvolvimento do jogo-arte,

working out compound strategies based on all the tricks it had been taught, and the new approaches were often much cleverer than those we would have tried.

tanto no plano estético quanto no plano desportivo. Estabelecendo um cenário onde os jogadores criam e as máquinas calculam, o xadrez avançado é emblemático de um profícuo modelo epistemológico, em que as pessoas fazem planos e emitem juízos, usando os computadores como ferramentas de análise. Ou, parafraseando o título de Weizenbaum (1976): Um modelo epistemológico (dentre tantos outros), onde o computador tem o poder de cálculo, e o ser humano, o dom da razão.¹²⁵

¹²⁵ Mas, entre o juízo e o cálculo, o elo da educação é indispensável. Apenas um jogador com elevada cultura enxadrística pode fazer o melhor uso de um computador, como ferramenta de apoio à sua criatividade.

6. Conclusão

Há cerca de doze anos, Hsu e seus colaboradores (1990) trabalhavam ativamente para multiplicar por mil o poder de cálculo do Deep Thought (antecessor do Deep Blue), com vistas a um posterior confronto com Garry Kasparov. Sobre as possibilidades da máquina, eles disseram que

Se a relação observada entre poder de processamento e força de jogo se mantiver, a máquina de próxima geração jogará em um nível de [rating performance] 3400, cerca de 800 pontos acima do Deep Thought atual, e 500 pontos acima do rating recorde de Kasparov¹²⁶ (p.24).

Porém, de acordo com a tabela usada pela FIDE para cálculo de rating, uma diferença de 500 pontos no rating entre dois jogadores significa que o jogador mais forte ganharia, ao longo do tempo, cerca de 96% dos pontos disputados como jogador mais fraco. Ou seja, para materializar a previsão de Hsu, a “próxima geração” do Deep Thought (no caso, o Deep Blue), deveria ter ganhado 96% dos pontos disputados contra Kasparov.

As ocorrências se deram de modo bem diverso. Em 1996, Kasparov venceu o match por 4 a 2, e em 1997, perdeu por 3,5 a 2,5. Assim, somados os pontos das duas competições, o score é de 6,5 a 5,5 (54% - 46%), favorável a Kasparov, o que significa uma diferença de rating performance de cerca de 30 pontos, a favor do ex-campeão mundial. Mesmo considerando que o Deep Blue não era mil vezes mais rápido do que o Deep Thought, mas *apenas* cerca de duzentas vezes mais rápido, essa diferença seria suficiente para superar Kasparov, se a relação entre velocidade de cálculo e força do jogo continuasse crescendo da forma observada por Hsu (1990, p.20-21).

No que tange à capacidade das máquinas para simular tarefas intelectuais, discrepâncias como estas, entre previsões e realizações, são comuns, conforme assinala Dreyfus (1992). Tais discrepâncias, costumam cair no esquecimento, ao passo em que os sucessos obtidos pelos pesquisadores são amplamente divulgados (o Deep Blue, afinal, venceu Kasparov!). Porém, a propaganda e as previsões otimistas em torno da inteligência artificial não escondem o fato de que a simulação da razão humana, por meio da força bruta computacional, está muito longe de ser conseguida – se é que pode

¹²⁶ If the observed relation between processing power and playing strength holds, the next-generation machine will play a 3400 [performance rating] level, about 800 points above today's Deep Thought and 500 points above Kasparov's rating record.

ser conseguida. A esse respeito, é Dreyfus (op.cit.) quem se exprime com maior clareza, antecipando uma lição que se aplica ao xadrez computacional, mesmo depois da vitória sobre Kasparov:

Resultados negativos, supondo-se que nós os reconheçamos como tais, podem ser interessantes. Resultados mais modestos, ao invés dos sucessos previstos, em ritmo acelerado, talvez indiquem algum fenômeno inesperado. Talvez estejamos nos defrontando com um *continuum* como o da velocidade, em que a aceleração crescente requer mais e mais energia quando nos aproximamos da velocidade da luz, ou talvez estejamos diante de uma descontinuidade, que requer não um esforço maior, mas técnicas completamente diferentes, como no caso do escalador de árvores que tenta alcançar a lua¹²⁷ (p.100).

O aspecto competitivo do xadrez computacional monopoliza a atenção dos observadores. Por isso, é importante ressaltar que as vitórias dos computadores, diante de grandes mestres humanos, são consequência direta dos milhares de horas que engenheiros, programadores e enxadristas de talento têm dedicado ao aperfeiçoamento das máquinas vitoriosas. Em outras palavras, a performance notável dos programas reflete, acima de tudo, o trabalho diligente dos seus criadores. Sob esta ótica, é interessante pensar que os criadores dos melhores programas conseguem, com a ajuda dos seus computadores, jogar com a eficácia de grandes mestres – embora não pensem como grandes mestres.

A diferença entre inteligência enxadrística e eficácia no jogo competitivo é há muito reconhecida por John McCarthy. Em meados de 1968, ele (juntamente com outros pesquisadores) apostou que, em dez anos, um programa de xadrez seria capaz de vencer um match contra o MI David Levy. Em 1978, ano-limite da aposta, Levy venceu com facilidade o desafio final, contra o programa CHESS 4.7. O Prof. McCarthy pagou a Levy as £250,00 devidas, e comentou que “estava contente que Levy não tivesse perdido para um programa de força bruta, porque, ao fazer a aposta, ele (McCarthy) supôs que a derrota do humano resultaria de uma ‘inteligência’ genuína do programa

¹²⁷ “Negative results, provided one recognizes them as such, can be interesting. Diminishing achievement, instead of predicted accelerating success, perhaps indicates some unexpected phenomenon. Perhaps we are pushing out on a continuum like that of velocity, where further acceleration costs more and more energy as we approach the speed of light, or perhaps we are instead facing a discontinuity, which requires not greater effort but entirely different techniques, as in the case of the tree-climbing man who tries to reach the moon”.

vencedor (Levy, 1991, p.101-102)”. Mais recentemente, McCarthy (2002b) sugeriu a organização de competições que só admitissem computadores com limites severos para o cálculo de variantes, de forma que a força bruta deixasse de ser um fator tão relevante da força das máquinas. Com isso, os programadores seriam forçados criar novas técnicas de avaliação posicional e seleção de variantes, tentando aproximar os programas da forma de pensar dos seres humanos.

A crítica de McCarthy se justifica, na medida em que a abordagem qualitativa do ser humano, no xadrez, tem proporcionado soluções que escapam à capacidade de cálculo dos melhores computadores. Não fica claro, contudo, como os programas de xadrez podem ser tornados mais “inteligentes”. Como já vimos, a maestria no xadrez é bem mais do que o simples reconhecimento de padrões, e a competência dos grande-mestres está fortemente calcada na sua capacidade de visualizar transformações posicionais, sem a necessidade de calcular variantes. Talvez essa habilidade, entre outras, possa ser imitada pelos computadores, mas não há evidências bastante fortes a favor dessa hipótese. A história do xadrez computacional, até agora, depõe contra ela.

O estímulo ao desenvolvimento de programas de xadrez cada vez mais fortes têm sido constante, nos últimos trinta anos. Nesse período, todos os programas que têm se destacado, tanto em competições contra seres humanos, como em competições entre máquinas, são programas baseados na força bruta, usando funções de avaliação heurística para seleção de variantes. Por outro lado, não se tem notícia de programas bem-sucedidos, baseados na imitação da inteligência humana, ou capazes de aprender por si mesmos. É possível que isso se deva, como quer McCarthy (2002a), a uma abordagem estritamente pragmática do xadrez computacional, voltada para o jogo de competição. Mas também é possível, como quer Dreyfus (1992), que a imitação da inteligência, mesmo em um domínio formal como o xadrez, não possa se concretizar através da computação eletrônica digital.

Na presente dissertação, a questão da eficácia e competência das máquinas de xadrez foi abordada apenas em termos do jogo competitivo, em que a força bruta, no cálculo de variantes, proporciona resultados expressivos. Mesmo assim, ficaram patentes as dificuldades que os programadores enfrentam para fazer face à inteligência dos melhores enxadristas do mundo: o Deep Blue, resultado de quinze anos de trabalho, a um custo aproximado de cinco milhões de dólares, venceu mas não convenceu. Aparentemente, o aumento da performance competitiva dos programas deixa de ser proporcional à profundidade de cálculo de variantes, a partir de um determinado nível de jogo. Além disso, a discrepância entre a notável performance média dos programas

de xadrez e seu jogo inexpressivo, em certas situações, demonstra que, apesar de eficazes, eles não são competentes. Em sentido estrito, as máquinas não jogam xadrez, apenas simulam o ato de jogar bem o bastante para vencer a maioria dos seres humanos.

As limitações metodológicas do xadrez computacional seriam ainda mais evidentes se optássemos por uma crítica mais radical, perguntando o que um programa de xadrez poderia fazer, além de jogar bem o xadrez: Ensinar finais? Aprender novas estratégias? Argumentar sobre uma posição? Justificar a introdução de novas idéias em uma abertura? E se após décadas de pesquisa, nenhuma dessas perguntas pode ser respondida afirmativamente, no domínio circunscrito do xadrez, por que seríamos tentados a crer que a inteligência artificial estaria próxima de conquistas muito maiores, com a criação de máquinas verdadeiramente “inteligentes”?

O fato de que as máquinas superam os seres humanos, em campos específicos de ação, tem sido tomado, erroneamente, como um indício de que elas podem superá-los em qualquer atividade. Esta é, por exemplo, uma das estratégias argumentativas usadas por Kurzweil (2000), para justificar seu otimismo sobre o que os computadores são capazes de fazer. O livro de Kurzweil anuncia um *admirável mundo novo*, capitaneado por máquinas superinteligentes. É uma obra tão ingênua, filosoficamente, que talvez não merecesse destaque em um debate sério, sobre as possibilidades e limites da inteligência artificial. Mas é também um best-seller, promovido por um sistema editorial poderoso, que inclui o elogio entusiasmado de personalidades e periódicos respeitáveis¹²⁸.

O computador é uma máquina única, que tem viabilizado realizações notáveis, em diversos campos da atividade humana. Mas é também objeto de uma pesada propaganda, a serviço de interesses ideológicos e econômicos muito fortes. É nesse

¹²⁸ “The Age of Spiritual Machines”, traz, na contracapa os elogios de Bill Gates e Marvin Minsky, entre outros. Este último, hiperbólico, assim se expressa: “[Kurzweil] links the relentless growth of our future technology to a universe in which Artificial Intelligence and Nanotechnology combine to bring unimaginable wealth and longevity, not merely to our descendants, but to some of those living today.” [Kurzweil associa o crescimento inexorável da nossa tecnologia futura com um universo em que a inteligência artificial e a nanotecnologia se combinam para nos trazer riqueza e longevidade inimagináveis, não apenas para os nossos descendentes, mas para algumas das pessoas que estão vivas hoje]. Em certo sentido, alguns barões da tecnologia já controlam “riquezas inimagináveis”, hoje me dia... Mas é de se questionar se a “riqueza e longevidade inimagináveis”, de que fala Minsky, seriam acompanhadas por uma justiça inimaginável para todos os povos, e de soluções para os inimagináveis problemas sociais e psicológicos que povoariam um mundo de seres híbridos (meio homens, meio computadores). Entretanto, este é o tipo de pergunta que Kurzweil simplesmente ignora.

contexto que deve ser visto o livro de Kurzweil, bem como outros livros e artigos na mesma linha. Ao fazer o elogio da máquina, a comunidade científica e o mundo empresarial passam a mensagem subliminar de que, quando as máquinas assumem o controle, as pessoas não têm com o que se preocupar. Na prática, porém, as máquinas nunca assumem o controle da ordem social. Os sistemas de informação de grande porte, instalados pelo mundo afora, continuam sendo projetados, implantados e operados atendendo aos interesses do grande capital, nos moldes vigentes há cinquenta anos. Ou seja: enquanto a redenção da humanidade pela via computacional não vem, os grupos de interesse estatais e privados prosseguem usando os computadores em seu próprio benefício – uma posição incômoda, para as regiões do planeta ainda carentes de eletrificação, educação fundamental e cuidados médicos elementares.

Com o advento dos computadores, a ordem institucional muda forçosamente na superfície, mas não necessariamente na essência. Atento a essa realidade, Weizenbaum (1976) já notava que

O advento da revolução computacional e o início da era dos computadores têm sido anunciados várias vezes. Mas se o triunfo de uma revolução deve ser medido em termos da profundidade das revisões que ela gera em torno das questões sociais, então não houve revolução computacional¹²⁹ (p.32).

Portanto, sob uma perspectiva filosófica, não há porque aceitar as promessas da inteligência artificial, quando atribui aos computadores a capacidade de realizar, por si próprios, transformações muito maiores do que as que o ser humano tem realizado com a ajuda deles. Como pensar em máquinas autônomas, engendrando transformações sociais em um futuro breve, quando as máquinas atuais não podem sequer lidar com as sutilezas do jogo de xadrez? Em meio às tantas áreas do conhecimento, seria a inteligência artificial a única imune à crítica, ao repetir-se em previsões frustradas, ignorando as etapas lógicas do desenvolvimento científico? Antes mesmo do problema moral, concernente ao que as máquinas deveriam fazer, deve-se enfrentar o problema epistemológico, concernente ao que as máquinas tem condições de fazer.

O presente texto não se deteve no aspecto moral do uso dos computadores, nem propôs limites definitivos para aquilo de que os computadores eletrônicos digitais são capazes. Mas, ao elucidar certas dificuldades para se superar a performance humana, e sobretudo

¹²⁹ The arrival of the Computer Revolution and the founding of the Computer Age have been announced many times. But if the triumph of a revolution is to be measured in terms of the profundity of the social revisions it entrained, then there has been no computer revolution.

para se imitar o pensamento humano, no jogo de xadrez, pretendeu enriquecer o debate sobre o que a inteligência artificial pode prometer. Em contrapartida, os sucessos do xadrez computacional foram igualmente levados em conta, como indício de que o computador é, por excelência, uma ferramenta epistemológica, cujo pleno potencial ainda está por ser revelado.

Em 1969, após uma aula de Karl Popper na Universidade de Emory, EUA, um dos assistentes perguntou se os computadores são “coisas pensantes”, capazes de criar novas teorias. A resposta de Popper foi clara:

Não. Também eles são produzidos pelo homem. Têm grande interesse e importância, mas não devem ser sobreestimados. Antes da era do computador, Einstein dissera: ‘O meu lápis é mais inteligente do que eu’ – uma frase assaz clara: não usaria o lápis, se este fosse mais inteligente do que ele. Ora, eu diria que um computador não é mais do que um lápis endeusado. Portanto, derrubemos do pedestal não só o homem como também, para variar, o computador (Popper, 1997, p.37).

Popper matou a charada; o computador é uma lápis programável. Usá-lo cada vez melhor, sem endeusá-lo jamais: eis um desafio para a educação, na entrada do século XXI.

Apêndice I: Confrontos Homem-Máquina

As partidas que se seguem, disputadas entre programas de computador e seres humanos, foram selecionadas por um motivo comum. Em todas elas, fica aparente a incapacidade das máquinas para elaborar e implementar planos de jogo em conformidade com as características de cada posição. Naturalmente, há muitas partidas em que o poder de cálculo das máquinas, contrapondo-se à imprecisão humana, proporciona vitórias em favor dos programas. Elas não foram selecionadas porque o que está em discussão, neste texto, são os limites epistemológicos da máquina, e não os limites psicológicos dos humanos. Os limites físicos, cognitivos e emocionais dos enxadristas se manifestam igualmente em partidas entre humanos, e os jogos contra computadores apenas radicalizam a questão.

As grandes partidas de xadrez têm sempre um *fio condutor*, que se traduz na concretização gradativa de propósitos definidos, através de recursos técnicos sofisticados. No jogo de um computador, os bons lances decorrem de uma confluência feliz entre força bruta e heurística, nunca de uma intencionalidade, que simplesmente não existe. Por isso, os seres humanos freqüentemente jogam vários lances bons em seguida, enquanto os computadores alternam lances bons com outros sem sentido, principalmente nas posições que requerem planejamento a longo prazo. Assim, o jogo sem propósito dos computadores ainda sofre revezes, quando defrontado por jogadores de elite em sua melhor forma. As disputas entre humanos e computadores não são apenas jogadas de marketing, mas oportunidades para que os programadores de xadrez identifiquem problemas que os milhares de jogos entre computadores não deixam transparecer.

Partida 1: MI David Levy x CHESS 4.7, agosto de 1978

1.c4 Cf6 2.a3 Cc6?! 3.Cc3 d5 4.cxd5 Cxd5 5.d3 Cxc3?!

Justamente o lance que David Levy queria provocar. O programa avalia positivamente o aparente enfraquecimento dos peões adversários, mas as Brancas obtém o domínio cômodo da casa d4, bem como uma via de ataque através da coluna "b".

6.bxc3 e5 7.g3

Para combinar a ação do bispo e da torre sobre b7.

7...Be7 8.Bg2 Dd6?

Ao programa, "parece" que a dama está em uma posição ativa, no centro do tabuleiro. Na verdade, ela será objeto de assédio por parte das peças adversárias, que ocuparão

boas posições com ganho de tempo. (Os programas modernos ainda cometem erros no que tange ao posicionamento da dama, ainda que de forma mais sutil.)

9.Cf3 Be6

Se a dama estivesse em d7, ao invés de d6, este lance ameaçaria com a troca do forte Bg2 branco.

10.0-0 0-0 11.Da4 Dc5?

Mais uma injustificável perda de tempo.

12.Bd2 b5

Este lance é criticado por Levy, porque "enfraquece a estrutura de peões sem provocação suficiente". Mas, tendo em vista a pressão iminente das Brancas sobre os pontos c6 e b7, este lance tem o mérito de afastar a dama branca, muito bem colocada, de modo que as peças negras possam se reorganizar. Contudo, o computador não tem nada disso "em mente": os erros posicionais graves ainda estão por vir.

13.Dc2 f6

Lance correto, que defende o peão e5 e evita um provável Cg5.

14.Tfb1 Tad8?!

Este lance esconde ameaças táticas no centro, mas as Brancas as evitam facilmente. As Negras deveriam reagrupar as suas peças para defender a ala da dama.

15.Db2 Tb8

A torre vai para a casa correta, após perder um tempo importante.

16.Be3

O bispo se coloca na casa ideal, com ganho de tempo, já que a dama é forçada a recuar do inútil décimo primeiro lance.

16...Dd6 17.Cd2 Bd5 18.Bxd5+ Dxd5 19.Db3 Dxb3 20.Cxb3 f5 21.Bc5 Bd6 22.Tb2 Rh8??

Lance absurdo, que nenhum programa moderno faria, pois afasta o rei do campo de batalha, na transição para o final. Segundo Levy, o CHESS 4.7 avalia que ainda está no meio-jogo e, na ausência de um plano concreto, segue o princípio de "colocar o rei em segurança".

23.Tab1 a6 24.Bxd6 cxd6 25.Cd2

David Levy faz uma avaliação otimista quanto às suas possibilidades de vitória. Porém, com jogo correto de ambas as partes, esta posição provavelmente levaria a um empate. Entretanto...

25...f4?

Com esse lance, o programa abre espaço para que o cavalo branco participe ativamente do jogo, através da casa e4. Em um final com cavalos, o enfraquecimento das casas centrais é muito grave.

26.Rg2 fxg3?

Evidentemente, falta um fio condutor. Este lance anula até mesmo as remotas possibilidades de agressão à ala do rei branca, a fim de compensar a fraqueza das

Negras na ala da dama. Além disso, as Negras ficam com os peões repartidos em três grupos, onerando a sua defesa (Um programa moderno dificilmente faria os dois últimos lances das Negras, tão ostensivamente antiposicionais). Talvez ainda houvesse tempo para os procedimentos defensivos adequados.

27.hxg3 Tbd8 28.a4

Agora, as Brancas forçam o enfraquecimento definitivo da ala da dama negra, obtendo uma vantagem decisiva.

28...Ca7 29.Ce4 bxa4 30.Tb6 d5 31.Cc5 Cb5 32.Cxa4 Ta8 33.c4

David Levy comenta que os computadores não avaliam adequadamente a força dos peões passados. Quase vinte anos depois dessa partida, o Deep Blue foi derrotado por Kasparov, no primeiro jogo do match de 1997, porque subestimou um par de peões passados do adversário. Atualmente, os programas comerciais melhoraram muito, nesse aspecto.

33...dxc4 34.dxc4 Cd4 35.e3 Cf3 36.c5 Cg5 37.c6 Ce4 38.c7 Txf2+ 39.Rg1 Tff8 40.Tb8 h5 41.Txa8 Txa8 42.Tb8+ 1-0

PARTIDA 2: MI G. Russek x Rebel Century, 08/01/2000

1.e3 d5 2.c4 dxc4 3.Bxc4 Cc6 4.d4 e5 5.Cf3 exd4 6.Cxd4 Ce5 7.Be2 c5 8.Cb5 Bf5 9.Dxd8+ Txd8 10.0-0 a6

As Negras saem da abertura com vantagem, devido à maioria de peões na ala da dama, controle do centro e maior atividade das suas peças. Nas mãos de um enxadrista de elite, a posição das Negras provavelmente poderia ser considerada vitoriosa.

11.C5c3 Cf6 12.Cd2 Bd3?!

O lance 12.Cd3 dava às Negras o par de bispos, com boas chances de vitória.

13.Bxd3 Cxd3 14.Cf3 Be7 15.a4 0-0 16.b3 Cg4

Ameaçando 17...Cgxf2, ganhando pelo menos um peão. Porém, as Brancas têm possibilidades de defesa. Interessante seria 16...Cxc1, seguido de 17...Td7, mantendo o bispo e a pressão na coluna "d". O Rebel parece não valorizar os bispos nesse tipo de posição, tanto quanto um jogador experiente.

17.Ta2 Bf6?!

A posição das Negras é dominante, mas esse lance permitirá a troca do bispo pelo cavalo. Sem o bispo, aumentam as dificuldades para avançar a maioria de peões na ala da dama. 17...f5 controlaria a casa e4, de modo que o subsequente ...Bf6 não daria às Brancas a oportunidade de trocar seu cavalo pelo bispo negro, como aconteceu no jogo. 17...Cxc1, seguido de 18...Td7 é uma alternativa, aproveitando a desunião temporária das torres brancas, a fim de consolidar o domínio da coluna "d".

18.Ce4 Tfe8 19.Cxf6+ Cxf6 20.Tc2 Ce4 21.Ba3 b5 22.axb5 axb5 23.Cd2 Cf6 24.Tb1 Cg4 25.Cf3 b4 26.Bb2 f6 27.h3 Cgxf2

Não há nada melhor. A vantagem das Negras desapareceu, e o lance do texto força uma posição de empate.

28.Cxf2 Cxf2 29.Rxf2 Td3 30.Bc1 Tc3 31.Bd2 Tc2 32.Re2 Td8 33.Rd1 Tc3 34.Re2 Tc2 35.Rd1 Tc3

Um impasse: as Brancas estão imobilizadas na defesa de seus pontos fracos, mas as Negras não têm como intensificar o ataque.

36.Re2 Tc2 ½-½

Um final apropriado, para uma partida em que o humano apenas se defendeu, e programa deixou de explorar diversas vias para materializar sua vantagem posicional.

PARTIDA 3: Junior 6 x GM Jeroen Piket, 15/07/2000

1.e4 g6 2.d4 Bg7 3.Cc3 c6 4.Cf3 d5 5.h3 a6 6.Bf4?!

A decisão sobre o posicionamento do BD deveria aguardar o desenrolar do jogo. Em f4, ele não cumpre nenhuma função estratégica relevante, além de obstruir um possível avanço dos peões brancos na ala do rei.

6...Cf6 7.e5 Cfd7 8.Dd2 e6 9.Bg5 Db6 10.0-0-0?!

O grande roque é um sinal de que o computador não "entende" a posição. A partir de agora, as Negras conquistam a iniciativa, através de um ataque ao rei adversário.

10...h6 11.Be3 Dc7 12.h4?

Lance sem sentido, que não contribui para a defesa e nem para o ataque. Um jogador experiente, com as Brancas, tentaria abrir o centro rapidamente. 12.Ce1, seguido de f4 e Cd3 é uma idéia que merece atenção. Claramente, o computador não tem um plano de jogo definido, na ausência de ameaças táticas imediatas.

12...b5 13.Bf4

Em treze lances, este é o quarto movimento desse bispo, sem qualquer proveito para as Brancas.

13...Cb6 14.a3?!

Enfraquece o roque precocemente, e não evita o avanço b4, que acabará ocorrendo, cedo ou tarde.

14...C8d7 15.Rb1 a5

Enquanto as Brancas não têm chances reais de ataque, na ala do rei, a iniciativa das Negras na ala da dama cresce continuamente.

16.Ca2 Da7 17.g4 Bf8 18.c3 Ba6 19.De1 Cc4 20.Bd2 Be7 21.Cc1 Cdb6

Nesse ponto, o ataque de Piket já é decisivo. O rei branco sofre constantes ameaças, enquanto o rei negro está em segurança, protegido pelo centro bloqueado.

22.h5 g5 23.Ca2 Rd7 24.Bc1 Thb8

Agora, todas as peças negras participam do ataque. As peças brancas, mal colocadas, não terão tempo para socorrer o rei.

25.Ra1 b4 26.Cd2 Cxd2 27.Bxd2 Bxf1 28.Txf1 Cc4 29.Tb1 b3 30.Cc1 Bxa3

As manobras táticas decisivas, com sacrifícios, são típicas das vitórias dos grandes mestres sobre os computadores.

**31.Dd1 [31.bxa3 b2+ 32.Ra2 bxc1D 33.Dxc1 Txb1 34.Rxb1 Tb8+ 35.Rc2 Db6-+]
31...Db6 32.bxa3 b2+ 33.Ra2 bxc1C+ 34.Dxc1 Dxb1+ 0-1**

PARTIDA 4: GM Loek van Wely x Rebel Century 4.0, 20/02/2002

1.d4 Cf6 2.c3

Lance típico do "jogo anticomputador". As Brancas não buscam a iniciativa; apenas reforçam o centro, aguardando o posicionamento das peças adversárias.

2...e6 3.Bg5 h6 4.Bh4 b6 5.Cd2 Bb7 6.e3 Be7 7.Cgf3 c5 8.Bd3 0-0 9.Qe2 d5?!

Esse lance restringe a ação do BD negro, e dá às Brancas acesso livre à casa e5, importante posto de ataque. 9.d6, mantendo o centro sob controle, merece atenção.

10.Ce5 Cc6 11.f4 Ce4 12.Bg3 Cxg3?

Caindo na armadilha posicional. O cavalo vale mais do que o bispo, nessa posição. Além disso, a captura abre a coluna h sobre o roque das Negras.

13.hxg3 Tc8?

Enquanto o ataque das Brancas ganha ímpeto, o computador faz um lance sem propósito. Um jogador experiente organizaria as defesas na ala do rei, ou tentaria uma ruptura central, imediatamente. 13...Cxe5, seguido de 14...d4 ou 14...f5, oferece contrachances, em vista da posição delicada do rei branco.

14.Cdf3 Bd6 15.g4 f6 16.Cg6 Tf7 17.g5 Tcc7 18.gxh6 gxh6 19.Txh6 Th7 20.Txh7 Txh7 21.0-0-0 c4 22.Bc2 Qe8 23.g4 Th6 24.Cfh4 Th7 25.Qg2 Cd8 26.Th1 b5 27.Cf3 Txh1+ 28.Qxh1 Cf7 29.Qh5 Qd7 30.Cgh4 Ch8 31.g5 f5 32.Ce5 Bxe5 33.dxe5 Bc8 34.Cf3 Qf7 35.Qh6 Qg6 36.Qxg6+ Cxg6

Finda a tempestade sobre o rei negro, as Brancas têm o saldo de um peão a mais, passado e defendido, e duas peças ativas. Em contrapartida, as Negras têm um bispo que praticamente não joga. A partida está decidida.

37.Cd4 Bd7 38.b3 Rg7 39.Bd1 Ce7 40.Rb2 Rg6 41.Be2 Cc8 42.a4 a6 43.axb5 axb5 44.Ra3

Há três ameaças simultâneas das Brancas, com as quais as Negras não podem lidar: a penetração do rei na ala da dama, o avanço do peão "g" e a entrada do bispo na ala do rei, via h5.

44...Ce7 45.Rb4 Cc6+ 46.Cxc6 Bxc6 47.Rc5 Be8 48.b4 1-0

PARTIDA 5: Deep Junior x GM Boris Gulko, 21/03/2002

1.e4 e6 2.d4 d5 3.Cd2 Cf6 4.e5 Cg8 5.Bd3 c5 6.c3 Cc6 7.Ce2 Bd7 8.0-0 Dc7 9.Cf3 c4 10.Bc2 h6 11.b3 Ca5 12.b4?! Cc6 13.b5?

Esse avanço de peão é uma chicotada no ar. Agora, o rei negro encontrará refúgio na ala da dama. As Brancas poderiam continuar com 13.Be3, Cd2 e f4, obtendo iniciativa na ala do rei, onde têm ampla vantagem de espaço.

13...Ca5 14.Tb1 0-0-0 15.h4 Rb8 16.h5 Ce7 17.Ba3 Cc8 18.Bb4 b6 19.Dd2 Cb7 20.a4 Be7 21.Tfd1 Tdf8 22.Te1 Thg8 23.De3

O primeiro, de uma série de lances sem objetivo definido, que tiram a dama branca de jogo por longo tempo.

23...Th8 24.Ch2 Be8 25.Dg3 Tfg8 26.Cf3 Ra8 27.Rh2 Dd8 28.Th1 Dc7 29.Ta1 Rb8 30.Tae1 Ra8 31.Dh3 Cd8 32.Tb1 Rb8 33.Ta1 Cb7 34.Dg3 Ra8 35.Tab1 Rb8 36.Rbf1 Cd8 37.Dh3 g5 38.Bxe7 Cxe7 39.Dg3

Os sucessivos movimentos da dama e das torres brancas, desvinculados de um plano específico, causam uma péssima impressão estética. É a máquina jogando sem um fio condutor.

39...Cf5 40.Dg4 Bd7 41.Cg3 Ce7 42.a5?!

Após esta ruptura, o peão a5 das Negras se torna perigoso, ao passo em que as Brancas não têm chances reais de ataque.

42...bxa5 43.Ta1 Tf8 44.Thf1 Cb7 45.Rg1 Cc8 46.Ch1 Cb6 47.Tfe1 Bxb5 48.Teb1 a6 49.Dh3 Ra7 50.Ch2 a4 51.Cg3 Ca5 52.Cg4 Cb3 53.Ta2 Cd7 54.Cf1 Bc6 55.Te1 Cb6 56.Df3 Cd7 57.Cfe3 Cd2 58.Dh3 Cb3 59.Df3 Cd2 60.Dd1 Cb3

Gulko opta pela repetição de jogadas, porque, para apoiar o avanço do peão a4, teria que descudar os seus pontos fracos na ala do rei.

61.Df3 ½-½

PARTIDA 6: GM Ilia Smirin x Deep Shredder, 15/04/2002

1.c4 e5 2.Cc3 Cf6 3.g3 Bb4

Lance normal nessa abertura, mas não é a melhor escolha para um computador, diante de um enxadrista de alto nível. Após uma provável troca do Bb5 pelo Cc3, o centro tende a se fechar, favorecendo o jogo humano.

4.Bg2 0-0 5.e4 Bxc3 6.bxc3 d6 7.Ce2 Bg4 8.f3 Be6

O Be6 e o Cf6 serão alvo dos peões brancos da ala do rei, permitindo que as Brancas ganhem um tempo precioso no ataque.

9.d3 c5?!

À esta altura, um jogador humano, com um pouco de experiência, iniciaria um plano para defender ala do rei, deixando em aberto a possibilidade de um contragolpe central, através do lance c6, oportunamente seguido de d5. Mas o programa, alheio ao iminente ataque das Brancas, prossegue fechando o centro, e com isso aumenta as chances de Smirin.

10.0-0 h6 11.h3 a6?

Ensaiaando um contragolpe característico (...b5). Porém, impõem-se procedimentos defensivos urgentes, como por exemplo, a manobra Ch7, seguido de f6 e Bf7. A posição

das Negras é inferior, mas uma defesa precisa talvez salvasse o jogo. O computador não se "apercebe" da força do ataque branco, até agora dissimulado. Enquanto Smirin age sistematicamente, de acordo com um plano de longa duração, o programa joga sem um fio condutor.

12.a4 b6 13.f4 Cc6 14.f5 Bd7 15.g4 Dc7?

O palco principal dos acontecimentos é a ala do rei, e a dama não tem função na casa c7. Provavelmente, as Negras já estão perdidas, depois desse lance.

16.Cg3 Ca5?

Este lance, e o próximo, preparam o avanço b5. Mas este é um procedimento demorado e inócuo, que não interfere no ataque decisivo das Brancas, na ala do rei.

17.h4

Sistematicamente, com lances simples, Smirin aumenta a pressão sobre o roque adversário.

17...Ch7 18.g5 hxg5 19.hxg5 Db7 20.Be3 b5 21.axb5 axb5 22.Dh5 bxc4 23.f6 Tfc8 24.fxg7 Bg4

Lance de aparência patética, para um ser humano, mas conforme a lógica dos programas, de retardar ao máximo o xeque-mate. Não há defesa contra as múltiplas ameaças brancas. [24...Be6 25.Tf6 cxd3 26.Cf5+-] 1-0

Apêndice II: Partidas do Deep Thought e do Deep Blue

Partida 1:

GM Bent Larsen x Deep Thought
American Open, Long Beach, 1990

1.c4 e5 2.g3 Cf6 3.Bg2 c6 4.Cf3 e4 5.Cd4 d5 6.cxd5 Dxd5 7.Cc2 Dh5 8.h4 Bf5 9.Ce3 Bc5 10.Db3 b6 11.Da4 0-0 12.Cc3 b5 13.Dc2 Bxe3 14.dxe3 Te8 15.a4 b4 16.Cb1 Cbd7 17.Cd2 Te6 18.b3 Td8 19.Bb2 Bg6 20.Cc4 Cd5 21.0-0-0 C7f6 22.Bh3 Bf5 23.Bxf5 Dxf5 24.f3 h5 25.Bd4 Td7 26.Rb2 Tc7 27.g4 hxg4 28.Thg1 c5 29.fxg4 Cxg4 30.Bxg7 Tg6 31.Dd2 Td7 32.Txg4 Txg4 33.Ce5 Cxe3 34.Dxd7 Cxd1+ 35.Dxd1 Tg3 36.Dd6 Rxg7 37.Cd7 Te3 38.Dh2 Rh7 39.Cf8+ Rh8 40.h5 Dd5 41.Cg6+ fxg6 42.hxg6+ Rg7 43.Dh7+ Rf6 0-1

Partida 2:

GM Robert Byrne x Deep Thought
Carnegie Mellon University, 1989

1.d4 d5 2.c4 dxc4 3.Cf3 Cf6 4.Da4+ c6 5.Dxc4 Bf5 6.Cc3 Cbd7 7.g3 Cb6 8.Db3 Be6 9.Dd1 Cbd5 10.Bg2 Cxc3 11.bxc3 Da5 12.Dc2 Ce4 13.Bb2 Df5 14.Da4 a6 15.0-0 Dh5 16.Tfe1 Df5 17.c4 f6 18.Db4 0-0-0 19.Tac1 Bf7 20.Db6 g5 21.d5 c5 22.Bc3 e5 23.dxe6 Bxe6 24.Tb1 Td7 25.Cd2 Cxc3 26.Bxb7+ Txb7 27.Dc6+ Rd8 28.Txb7 Bd7 29.Txd7+ Dxd7 30.Dxf6+ Re8 31.Dxc3 Bg7 32.De3+ De7 33.Ce4 Bd4 34.Dd3 Tf8 35.e3 Be5 36.Dd5 Tf7 37.Cxc5 Dd6 38.Da8+ Re7 39.Ce4 Dd3 40.Cxg5 Tf5 41.Dg8 1-0

Partida 3:

Mephisto x Deep Thought
20th ACM Championship, Reno, 1989

1.d4 d5 2.c4 dxc4 3.Cf3 Cf6 4.e3 Bg4 5.Bxc4 e6 6.h3 Bh5 7.Cc3 Cbd7 8.g4 Bg6 9.Ch4 Be4 10.Cxe4 Cxe4 11.Cf3 Cd6 12.Bb3 De7 13.Bd2 h5 14.Tg1 hxg4 15.hxg4 0-0-0 16.Ba5 b6 17.Bb4 a5 18.Bxd6 Dxd6 19.Dc2 Be7 20.0-0-0 Th3 21.Cd2 c6 22.Th1 Tdh8 23.Txh3 Txh3 24.Ce4 Dc7 25.Rb1 g5 26.Tc1 Rb7 27.Ba4 Cb8 28.Cd2 Dd7 29.Bb3 Ca6 30.De4 Cb4 31.a3 Cd5 32.Dg2 Th8 33.Ce4 f6 34.Dg3 Tg8 35.Th1 f5 36.gxf5 exf5 37.Dh3 Tf8 38.Cd2 Bf6 39.Dh7 Tf7 40.Dh6 De6 41.Dg6 Tg7 42.Th7 Txh7 43.Dxh7+ Be7 44.Rc1 Rc7 45.Cf3 Rd8 46.Ce5 g4 47.Dh8+ Rc7 48.Rd2 Rb7 49.Cxc6 Dxc6 50.De5 Cc7 51.Dxe7 Dg2 52.Dh4 f4 53.exf4 De4 54.Dxg4 Dxd4+ 55.Rc1 Dxf2 56.Df5 Df3 57.Rc2 Rc6 58.De5 Cd5 59.De6+ Rc5 60.Bxd5 Dxd5 61.Dxd5+ Rxd5 62.Rd3 a4 63.Rc3 Rc5 64.f5 1-0

Partida 4:
GM Garry Kasparov x Deep Thought
AGS Computer Challenge, 1989

1.d4 d5 2.c4 dxc4 3.e4 Cc6 4.Cf3 Bg4 5.d5 Ce5 6.Cc3 c6 7.Bf4 Cg6 8.Be3 cxd5 9.exd5
Ce5 10.Dd4 Cxf3+ 11.gxf3 Bxf3 12.Bxc4 Dd6 13.Cb5 Df6 14.Dc5 Db6 15.Da3 e6
16.Cc7+ Dxc7 17.Bb5+ Dc6 18.Bxc6+ bxc6 19.Bc5 Bxc5 20.Dxf3 Bb4+ 21.Re2 cxd5
22.Dg4 Be7 23.Thc1 Rf8 24.Tc7 Bd6 25.Tb7 Cf6 26.Da4 a5 27.Tc1 h6 28.Tc6 Ce8
29.b4 Bxh2 30.bxa5 Rg8 31.Db4 Bd6 32.Txd6 Cxd6 33.Tb8+ Txb8 34.Dxb8+ Rh7
35.Dxd6 Tc8 36.a4 Tc4 37.Dd7 1-0

Partida 5:
Deep Thought x GM Garry Kasparov
AGS Computer Challenge, 1989

1.e4 c5 2.c3 e6 3.d4 d5 4.exd5 exd5 5.Cf3 Bd6 6.Be3 c4 7.b3 cxb3 8.axb3 Ce7 9.Ca3
Cbc6 10.Cb5 Bb8 11.Bd3 Bf5 12.c4 0-0 13.Ta4 Dd7 14.Cc3 Bc7 15.Bxf5 Dxf5 16.Ch4
Dd7 17.0-0 Tad8 18.Te1 Tfe8 19.c5 Ba5 20.Dd3 a6 21.h3 Bxc3 22.Dxc3 Cf5 23.Cxf5
Dxf5 24.Ta2 Te6 25.Tae2 Tde8 26.Dd2 f6 27.Dc3 h5 28.b4 T8e7 29.Rh1 g5 30.Rg1 g4
31.h4 Te4 32.Db2 Ca7 33.Dd2 T4e6 34.Dc1 Cb5 35.Dd2 Ca3 36.Dd1 Rf7 37.Db3 Cc4
38.Rh2 Te4 39.g3 Df3 40.b5 a5 41.c6 f5 42.cxb7 Txb7 43.Rg1 f4 44.gxf4 g3 45.Dd1
Tbe7 46.b6 gxf2+ 47.Txf2 Dxd1 48.Txd1 Txe3 49.Tg2 Cxb6 50.Tg5 a4 51.Txh5 a3
52.Td2 Te2 0-1

Partida 6:
GM Anatoly Karpov x Deep Thought
Harvard University, 1990

1.e4 c6 2.d4 d5 3.Cd2 g6 4.c3 Bg7 5.e5 f6 6.f4 Ch6 7.Cgf3 0-0 8.Be2 fxe5 9.fxe5 c5
10.Cb3 cxd4 11.cxd4 Cc6 12.0-0 Db6 13.Rh1 a5 14.a4 Bf5 15.Bg5 Be4 16.Cc5 Dxb2
17.Cxe4 dxe4 18.Tb1 Da3 19.Bc1 Dc3 20.Bd2 Da3 21.Bc1 Dc3 22.Tb3 Da1 23.Bc4+
Rh8 24.Bxh6 Dxd1 25.Bxg7+ Rxc7 26.Txd1 exf3 27.gxf3 Ta7 28.Bd5 Td8 29.Tb5 Ta6
30.Bc4 Ta7 31.Bd5 Ta6 32.Tc5 Td7 33.Rg2 Tb6 34.Bxc6 bxc6 35.Rf2 Td5 36.Txd5
cxd5 37.Tc1 Tb4 38.Re3 Txa4 39.Tc5 e6 40.Tc7+ Rg8 41.Te7 Ta3+ 42.Rf4 Td3
43.Txe6 Txd4+ 44.Rg5 Rf7 45.Ta6 a4 46.f4 h6+ 47.Rg4 Tc4 48.h4 Td4 49.Tf6+ Rg7
50.Ta6 Rf7 51.h5 gxh5+ 52.Rf5 Rg7 53.Ta7+ Rf8 54.e6 Te4 55.Td7 Tc4 56.Txd5 h4
57.Td3 Re7 58.Td7+ Rf8 59.Th7 h5 60.Re5 h3 61.f5 Rg8 62.Txh5 a3 63.Txh3 a2
64.Ta3 Tc5+ 65.Rf6 1-0

Partida 7:
GM David Bronstein x Deep Thought
New York, 1992

1.e4 c5 2.b4 cxb4 3.a3 d5 4.exd5 Dxd5 5.Cf3 Bg4 6.axb4 De4+ 7.Be2 Bxf3 8.gxf3
Dxb4 9.Ca3 Da5 10.Bb2 Cc6 11.c4 Dg5 12.Db3 0-0-0 13.d4 Dg2 14.0-0-0 Dxf2 15.d5
Ca5 16.Db5 De3+ 17.Rb1 Dxe2 18.Dxa5 a6 19.The1 Df2 20.c5 Cf6 21.Be5 Cxd5
22.Cc4 Dxe1 23.Txe1 f6 24.Cb6+ Cxb6 25.Dxb6 1-0

Partida 8:**Deep Blue x GM Garry Kasparov
Philadelphia, 1996, 1ª partida do match**

1.e4 c5 2.c3 d5 3.exd5 Dxd5 4.d4 Cf6 5.Cf3 Bg4 6.Be2 e6 7.h3 Bh5 8.0-0 Cc6 9.Be3
cxd4 10.cxd4 Bb4 11.a3 Ba5 12.Cc3 Dd6 13.Cb5 De7 14.Ce5 Bxe2 15.Dxe2 0-0
16.Tac1 Tac8 17.Bg5 Bb6 18.Bxf6 gxf6 19.Cc4 Tfd8 20.Cxb6 axb6 21.Tfd1 f5 22.De3
Df6 23.d5 Txd5 24.Txd5 exd5 25.b3 Rh8 26.Dxb6 Tg8 27.Dc5 d4 28.Cd6 f4 29.Cxb7
Ce5 30.Dd5 f3 31.g3 Cd3 32.Tc7 Te8 33.Cd6 Te1+ 34.Rh2 Cxf2 35.Cxf7+ Rg7
36.Cg5+ Rh6 37.Txh7+ 1-0

Partida 9:**GM Garry Kasparov Garry x Deep Blue
Philadelphia, 1996, 2ª partida do match**

1.Cf3 d5 2.d4 e6 3.g3 c5 4.Bg2 Cc6 5.0-0 Cf6 6.c4 dxc4 7.Ce5 Bd7 8.Ca3 cxd4
9.Caxc4 Bc5 10.Db3 0-0 11.Dxb7 Cxe5 12.Cxe5 Tb8 13.Df3 Bd6 14.Cc6 Bxc6
15.Dxc6 e5 16.Tb1 Tb6 17.Da4 Db8 18.Bg5 Be7 19.b4 Bxb4 20.Bxf6 gxf6 21.Dd7 Dc8
22.Dxa7 Tb8 23.Da4 Bc3 24.Txb8 Dxb8 25.Be4 Dc7 26.Da6 Rg7 27.Dd3 Tb8 28.Bxh7
Tb2 29.Be4 Txa2 30.h4 Dc8 31.Df3 Ta1 32.Txa1 Bxa1 33.Dh5 Dh8 34.Dg4+ Rf8
35.Dc8+ Rg7 36.Dg4+ Rf8 37.Bd5 Re7 38.Bc6 Rf8 39.Bd5 Re7 40.Df3 Bc3 41.Bc4
Dc8 42.Dd5 De6 43.Db5 Dd7 44.Dc5+ Dd6 45.Da7+ Dd7 46.Da8 Dc7 47.Da3+ Dd6
48.Da2 f5 49.Bxf7 e4 50.Bh5 Df6 51.Da3+ Rd7 52.Da7+ Rd8 53.Db8+ Rd7 54.Be8+
Re7 55.Bb5 Bd2 56.Dc7+ Rf8 57.Bc4 Bc3 58.Rg2 Be1 59.Rf1 Bc3 60.f4 exf3 61.exf3
Bd2 62.f4 Re8 63.Dc8+ Re7 64.Dc5+ Rd8 65.Bd3 Be3 66.Dxf5 Dc6 67.Df8+ Rc7
68.De7+ Rc8 69.Bf5+ Rb8 70.Dd8+ Rb7 71.Dd7+ Dxd7 72.Bxd7 Rc7 73.Bb5 1-0

Partida 10:**Deep Blue x GM Garry Kasparov
Philadelphia, 1996, 3ª partida do match**

1.e4 c5 2.c3 d5 3.exd5 Dxd5 4.d4 Cf6 5.Cf3 Bg4 6.Be2 e6 7.0-0 Cc6 8.Be3 cxd4 9.cxd4
Bb4 10.a3 Ba5 11.Cc3 Dd6 12.Ce5 Bxe2 13.Dxe2 Bxc3 14.bxc3 Cxe5 15.Bf4 Cf3+
16.Dxf3 Dd5 17.Dd3 Tc8 18.Tfc1 Dc4 19.Dxc4 Txc4 20.Tcb1 b6 21.Bb8 Ta4 22.Tb4
Ta5 23.Tc4 0-0 24.Bd6 Ta8 25.Tc6 b5 26.Rf1 Ta4 27.Tb1 a6 28.Re2 h5 29.Rd3 Td8
30.Be7 Td7 31.Bxf6 gxf6 32.Tb3 Rg7 33.Re3 e5 34.g3 exd4+ 35.cxd4 Te7+ 36.Rf3
Td7 37.Td3 Taxd4 38.Txd4 Txd4 39.Txa6 b4 ½-½

Partida 11:**GM Garry Kasparov x Deep Blue
Philadelphia, 1996, 4ª partida do match**

1.Cf3 d5 2.d4 c6 3.c4 e6 4.Cbd2 Cf6 5.e3 Cbd7 6.Bd3 Bd6 7.e4 dxe4 8.Cxe4 Cxe4
9.Bxe4 0-0 10.0-0 h6 11.Bc2 e5 12.Te1 exd4 13.Dxd4 Bc5 14.Dc3 a5 15.a3 Cf6 16.Be3
Bxe3 17.Txe3 Bg4 18.Ce5 Te8 19.Tae1 Be6 20.f4 Dc8 21.h3 b5 22.f5 Bxc4 23.Cxc4
bxc4 24.Txe8+ Cxe8 25.Te4 Cf6 26.Txc4 Cd5 27.De5 Dd7 28.Tg4 f6 29.Dd4 Rh7
30.Te4 Td8 31.Rh1 Dc7 32.Df2 Db8 33.Ba4 c5 34.Bc6 c4 35.Txc4 Cb4 36.Bf3 Cd3
37.Dh4 Dxb2 38.Dg3 Dxa3 39.Tc7 Df8 40.Ta7 Ce5 41.Txa5 Df7 42.Txe5 fxe5
43.Dxe5 Te8 44.Df4 Df6 45.Bh5 Tf8 46.Bg6+ Rh8 47.Dc7 Dd4 48.Rh2 Ta8 49.Bh5
Df6 50.Bg6 Tg8 ½-½

Partida 12:**Deep Blue x GM Garry Kasparov
Philadelphia, 1996, 5ª partida do match**

1.e4 e5 2.Cf3 Cf6 3.Cc3 Cc6 4.d4 exd4 5.Cxd4 Bb4 6.Cxc6 bxc6 7.Bd3 d5 8.exd5 cxd5
9.0-0 0-0 10.Bg5 c6 11.Df3 Be7 12.Tae1 Te8 13.Ce2 h6 14.Bf4 Bd6 15.Cd4 Bg4
16.Dg3 Bxf4 17.Dxf4 Db6 18.c4 Bd7 19.cxd5 cxd5 20.Txe8+ Txe8 21.Dd2 Ce4
22.Bxe4 dxe4 23.b3 Td8 24.Dc3 f5 25.Td1 Be6 26.De3 Bf7 27.Dc3 f4 28.Td2 Df6
29.g3 Td5 30.a3 Rh7 31.Rg2 De5 32.f3 e3 33.Td3 e2 34.gxf4 e1D 35.fxe5 Dxc3
36.Txc3 Txd4 37.b4 Bc4 38.Rf2 g5 39.Te3 Td2+ 40.Re1 Td3 41.Rf2 Rg6 42.Txd3
Bxd3 43.Re3 Bc2 44.Rd4 Rf5 45.Rd5 h5 0-1

Partida 13:**GM Garry Kasparov x Deep Blue
Philadelphia, 1996, 6ª partida do match**

1.Cf3 d5 2.d4 c6 3.c4 e6 4.Cbd2 Cf6 5.e3 c5 6.b3 Cc6 7.Bb2 cxd4 8.exd4 Be7 9.Tc1 0-
0 10.Bd3 Bd7 11.0-0 Ch5 12.Te1 Cf4 13.Bb1 Bd6 14.g3 Cg6 15.Ce5 Tc8 16.Cxd7
Dxd7 17.Cf3 Bb4 18.Te3 Tfd8 19.h4 Cge7 20.a3 Ba5 21.b4 Bc7 22.c5 Te8 23.Dd3 g6
24.Te2 Cf5 25.Bc3 h5 26.b5 Cce7 27.Bd2 Rg7 28.a4 Ta8 29.a5 a6 30.b6 Bb8 31.Bc2
Cc6 32.Ba4 Te7 33.Bc3 Ce5 34.dxe5 Dxa4 35.Cd4 Cxd4 36.Dxd4 Dd7 37.Bd2 Te8
38.Bg5 Tc8 39.Bf6+ Rh7 40.c6 bxc6 41.Dc5 Rh6 42.Tb2 Db7 43.Tb4 1-0

Partida 14:**GM Garry Kasparov x Deep Blue
New York, 1997, 1ª partida do match**

1.Cf3 d5 2.g3 Bg4 3.b3 Cd7 4.Bb2 e6 5.Bg2 Cgf6 6.0-0 c6 7.d3 Bd6 8.Cbd2 0-0 9.h3 Bh5 10.e3 h6 11.De1 Da5 12.a3 Bc7 13.Ch4 g5 14.Chf3 e5 15.e4 Tfe8 16.Ch2 Db6 17.Dc1 a5 18.Te1 Bd6 19.Cdf1 dxe4 20.dxe4 Bc5 21.Ce3 Tad8 22.Chf1 g4 23.hxg4 Cxg4 24.f3 Cxe3 25.Cxe3 Be7 26.Rh1 Bg5 27.Te2 a4 28.b4 f5 29.exf5 e4 30.f4 Bxe2 31.fxg5 Ce5 32.g6 Bf3 33.Bc3 Db5 34.Df1 Dxf1+ 35.Txf1 h5 36.Rg1 Rf8 37.Bh3 b5 38.Rf2 Rg7 39.g4 Rh6 40.Tg1 hxg4 41.Bxg4 Bxg4 42.Cxg4+ Cxg4+ 43.Txg4 Td5 44.f6 Td1 45.g7 1-0

Partida 15:**Deep Blue x GM Garry Kasparov
New York, 1997, 2ª partida do match**

1.e4 e5 2.Cf3 Cc6 3.Bb5 a6 4.Ba4 Cf6 5.0-0 Be7 6.Te1 b5 7.Bb3 d6 8.c3 0-0 9.h3 h6 10.d4 Te8 11.Cbd2 Bf8 12.Cf1 Bd7 13.Cg3 Ca5 14.Bc2 c5 15.b3 Cc6 16.d5 Ce7 17.Be3 Cg6 18.Dd2 Ch7 19.a4 Ch4 20.Cxh4 Dxh4 21.De2 Dd8 22.b4 Dc7 23.Tec1 c4 24.Ta3 Tec8 25.Tca1 Dd8 26.f4 Cf6 27.fxe5 dxe5 28.Df1 Ce8 29.Df2 Cd6 30.Bb6 De8 31.T3a2 Be7 32.Bc5 Bf8 33.Cf5 Bxf5 34.exf5 f6 35.Bxd6 Bxd6 36.axb5 axb5 37.Be4 Txa2 38.Dxa2 Dd7 39.Da7 Tc7 40.Db6 Tb7 41.Ta8+ Rf7 42.Da6 Dc7 43.Dc6 Db6+ 44.Rf1 Tb8 45.Ta6 1-0

Partida 16:**GM Garry Kasparov x Deep Blue
New York, 1997, 3ª partida do match**

1.d3 e5 2.Cf3 Cc6 3.c4 Cf6 4.a3 d6 5.Cc3 Be7 6.g3 0-0 7.Bg2 Be6 8.0-0 Dd7 9.Cg5 Bf5 10.e4 Bg4 11.f3 Bh5 12.Ch3 Cd4 13.Cf2 h6 14.Be3 c5 15.b4 b6 16.Tb1 Rh8 17.Tb2 a6 18.bxc5 bxc5 19.Bh3 Dc7 20.Bg4 Bg6 21.f4 exf4 22.gxf4 Da5 23.Bd2 Dxa3 24.Ta2 Db3 25.f5 Dxd1 26.Bxd1 Bh7 27.Ch3 Tfb8 28.Cf4 Bd8 29.Cfd5 Cc6 30.Bf4 Ce5 31.Ba4 Cxd5 32.Cxd5 a5 33.Bb5 Ta7 34.Rg2 g5 35.Bxe5+ dxe5 36.f6 Bg6 37.h4 gxh4 38.Rh3 Rg8 39.Rxh4 Rh7 40.Rg4 Bc7 41.Cxc7 Txc7 42.Txa5 Td8 43.Tf3 Rh8 44.Rh4 Rg8 45.Ta3 Rh8 46.Ta6 Rh7 47.Ta3 Rh8 48.Ta6 ½-½

Partida 17:**Deep Blue x GM Garry Kasparov
New York, 1997, 4ª partida do match**

1.e4 c6 2.d4 d6 3.Cf3 Cf6 4.Cc3 Bg4 5.h3 Bh5 6.Bd3 e6 7.De2 d5 8.Bg5 Be7 9.e5 Cfd7
10.Bxe7 Dxe7 11.g4 Bg6 12.Bxg6 hxg6 13.h4 Ca6 14.0-0-0 0-0-0 15.Tdg1 Cc7 16.Rb1
f6 17.exf6 Dxf6 18.Tg3 Tde8 19.Te1 Thf8 20.Cd1 e5! 21.dxe5 Df4 22.a3 Ce6 23.Cc3
Cdc5 24.b4! Cd7 25.Dd3 Df7 26.b5 Cdc5 27.De3 Df4 28.bxc6 bxc6 29.Td1 Rc7
30.Ra1 Dxe3 31.fxe3 Tf7 32.Th3 Tef8 33.Cd4 Tf2 34.Tb1 Tg2 35.Cce2 Txg4
36.Cxe6+ Cxe6 37.Cd4 Cxd4 38.exd4 Txd4 39.Tg1 Tc4 40.Txg6 Txc2 41.Txg7+ Rb6
42.Tb3+ Rc5 43.Txa7 Tf1+ 44.Tb1 Tff2 45.Tb4 Tc1+ 46.Tb1 Tcc2 47.Tb4 Tc1+
48.Tb1 Txb1+ 49.Rxb1 Te2 50.Te7 Th2 51.Th7 Rc4 52.Tc7 c5 53.e6 Txh4 54.e7 Te4
55.a4 Rb3 56.Rc1 ½-½

Partida 18:**GM Garry Kasparov x Deep Blue
New York, 1997, 5ª partida do match**

1.Cf3 d5 2.g3 Bg4 3.Bg2 Cd7 4.h3 Bxf3 5.Bxf3 c6 6.d3 e6 7.e4 Ce5 8.Bg2 dxe4 9.Bxe4
Cf6 10.Bg2 Bb4+ 11.Cd2 h5 12.De2 Dc7 13.c3 Be7 14.d4 Cg6 15.h4 e5 16.Cf3 exd4
17.Cxd4 0-0-0 18.Bg5 Cg4 19.0-0-0 The8 20.Dc2 Rb8 21.Rb1 Bxg5 22.hxg5 C6e5
23.The1 c5 24.Cf3 Txd1+ 25.Txd1 Cc4 26.Da4 Td8 27.Te1 Cb6 28.Dc2 Dd6 29.c4
Dg6 30.Dxg6 fxg6 31.b3 Cxf2 32.Te6 Rc7 33.Txg6 Td7 34.Ch4 Cc8 35.Bd5 Cd6
36.Te6 Cb5 37.cxb5 Txd5 38.Tg6 Td7 39.Cf5 Ce4 40.Cxg7 Td1+ 41.Rc2 Td2+ 42.Rc1
Txa2 43.Cxh5 Cd2 44.Cf4 Cxb3+ 45.Rb1 Td2 46.Te6 c4 47.Te3 Rb6 48.g6 Rxb5 49.g7
Rb4 ½-½

Partida 19:**Deep Blue x GM Garry Kasparov
New York, 1997, 6ª partida do match**

1.e4 c6 2.d4 d5 3.Cc3 dxe4 4.Cxe4 Cd7 5.Cg5 Cgf6 6.Bd3 e6 7.C1f3 h6 8.Cxe6 De7
9.0-0 fxe6 10.Bg6+ Rd8 11.Bf4 b5 12.a4 Bb7 13.Te1 Cd5 14.Bg3 Rc8 15.axb5 cxb5
16.Dd3 Bc6 17.Bf5 exf5 18.Txe7 Bxe7 19.c4 1-0

Apêndice III: Partidas do Hiarc 8

Partida 1: GM Boris Gulko x Hiarc 8, 19/03/2002

1.c4 e5 2.Cc3 Cf6 3.g3 d5 4.cxd5 Cxd5 5.Bg2 Cxc3 6.bxc3 Be7 7.Cf3 Cc6 8.0-0 Be6 9.d4 e4 10.Ce1 f5 11.f3 exf3 12.Bxf3 0-0 13.Cg2 Ca5 14.d5 Bd7 15.Dd3 Bc5+ 16.Be3 Bxe3+ 17.Cxe3 De7 18.Dd4 Tae8 19.Cg2 b6 20.Tae1 Cb7 21.e4 fxe4 22.Txe4 Dc5 23.Cf4 Txe4 24.Bxe4 Dxd4+ 25.cxd4 Bb5 26.Tc1 Cd6 27.Bc2 Bc4 28.Bb3 b5 29.Ce6 Tc8 30.Bxc4 bxc4 31.Rf2 Rf7 32.Re3 c6 33.Tf1+ Re7 34.Cxg7 cxd5 35.Cf5+ Cxf5+ 36.Txf5 Re6 37.Te5+ Rd6 38.Rd2 c3+ 39.Rc2 Tc4 40.Th5 Txd4 41.Rxc3 Tc4+ 42.Rd3 Ta4 43.Txh7 a5 44.h4 Txa2 45.g4 Ta3+ 46.Rc2 Tg3 47.g5 a4 48.Ta7 Tg2+ ½-½

Partida 2: Hiarc 8 x GM Boris Gulko, 25/03/2002

1.e4 d6 2.d4 Cf6 3.Cc3 g6 4.Cf3 Bg7 5.Be2 0-0 6.0-0 c5 7.d5 Ca6 8.Te1 Cc7 9.Bf4 b6 10.Dd2 Te8 11.a4 Bb7 12.Bc4 a6 13.Ta3 Tb8 14.Tb3 Ba8 15.e5 Ch5 16.Bh6 f5 17.Bxg7 Rxc7 18.Cg5 h6 19.exd6 exd6 20.Txe8 Dxe8 21.Ce6+ Cxe6 22.dxe6 Dd8 23.Bxa6 d5 24.a5 Dd6 25.Ca4 Cf6 26.Txb6 Txb6 27.Cxb6 Dxe6 28.De2 Dxe2 29.Bxe2 Bc6 30.Bf3 Bb7 31.c4 Ba6 32.Bxd5 Cxd5 33.cxd5 Bb5 34.Ca8 1-0

Partida 3: Hiarc 8 x GM Ilya Smirin, 15/04/2002

1.d4 Cf6 2.c4 g6 3.Cc3 Bg7 4.e4 d6 5.Cf3 0-0 6.Be2 e5 7.dxe5 dxe5 8.Dxd8 Txd8 9.Bg5 Te8 10.Cd5 Cxd5 11.cxd5 c6 12.Bc4 b5 13.Bb3 a5 14.a4 Bd7 15.Cd2 bxa4 16.Bxa4 cxd5 17.exd5 Bxa4 18.Txa4 Cd7 19.Be3 Bf8 20.Cc4 Bb4+ 21.Re2 Tec8 22.b3 Tab8 23.Cxa5 Bxa5 24.Txa5 Txb3 25.Td1 Rf8 26.d6 Tb7 27.Tc1 Txc1 28.Ta8+ Tb8 29.Txb8+ Cxb8 30.Bxc1 Re8 31.Rd3 Rd7 32.Ba3 f5 33.g3 g5 34.h3 h5 35.Rc4 Rc6 36.Rd3 Cd7 37.f3 Cf6 38.g4 fxg4 39.fxg4 hxg4 40.hxg4 ½-½

Partida 4: GM Ilya Smirin x Hiarc 8, 24/04/2002

1.c4 e5 2.Cf3 Cf6 3.g3 d5 4.cxd5 Cxd5 5.Bg2 Cc6 6.d4 cxd4 7.Cxd4 Cdb4 8.Cxc6 Dxd1+ 9.Rxd1 Cxc6 10.Cc3 Bd7 11.Be3 e5 12.Tc1 Td8 13.Re1 Be7 14.f4 exf4 15.gxf4 0-0 16.Cd5 Bd6 17.Rf2 Be6 18.Thd1 Td7 19.Cc3 Tfd8 20.a3 Ce7 21.Be4 Cf5 22.Bxf5 Bxf5 23.Td4 a6 24.Ca4 Be7 25.Txd7 Bxd7 26.Cc3 Bh4+ 27.Rg1 f5 28.Td1 Bc6 29.Txd8+ Bxd8 30.Bd4 Rf7 31.e3 h6 32.Ce2 g5 33.Rf2 Bc7 34.Bc3 Be4 35.Rg3 g4 36.Bd4 Rg6 37.Rf2 h5 38.Bc3 Bd8 39.Bd4 a5 40.Bc3 a4 41.Cd4 Bh4+ 42.Rg1 Be7 43.Rf2 Bd5 44.Rg3 h4+ 45.Rf2 Be4 46.Rg1 g3 47.h3 Bc5 48.Bb4 Bxb4 49.axb4 Rf7 50.Cb5 Re6 51.Cc3 Bc2 52.Rg2 Rd6 53.Rg1 Rc6 54.Rg2 b5 55.Rg1 Bd3 56.Rg2 Be4+ 57.Rg1 Bc2 58.Rg2 Bd3 59.Rg1 Be4 60.Rf1 ½-½

Apêndice IV: Sistema Algébrico de Notação

O sistema algébrico de notação atualmente é o mais utilizado para o registro de partidas de xadrez.

A idéia é muito simples: cada peça é representada por sua inicial, em maiúscula, de acordo com o idioma em que a notação é usada. Em Português,

R – Rei **D** – Dama **T** – Torre **B** – Bispo **C** – Cavalo

em inglês,

K – King **Q** – Queen **R** – Rook **B** – Bishop
N – Knight (**Kt**, em edições antigas)

e, em castelhano,

R – Rey **D** – Dama **T** – Torre **A** – Alfil **C** – Caballo

Há, ainda, publicações internacionais que substituem a inicial maiúscula por um ícone que representa a peça.

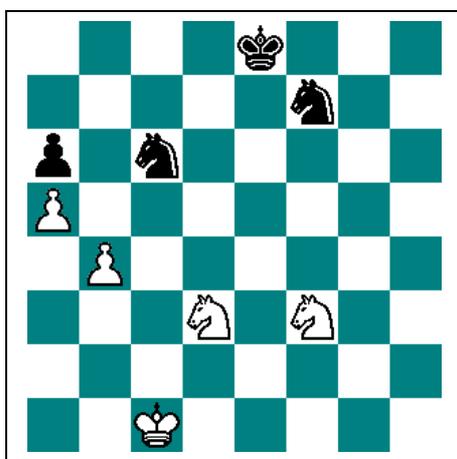
Devido à forma peculiar de movimentação dos peões, seus movimentos são dados apenas pelas coordenadas pertinentes, sem a inclusão de uma inicial maiúscula.

Para definição das coordenadas, as colunas do tabuleiro são marcadas com as letras de **a** a **h**, da esquerda para a direita, sob o ponto de vista do condutor das Brancas. Ainda sob esse ponto de vista, as horizontais são numeradas de **1** a **8**, de forma que cada casa tem uma coordenada, de **a1** a **h8**. A partir daí, basta determinar a peça jogada e a coordenada de destino, da forma mais abreviada possível. Para exprimir o movimento de um peão, basta escrever a coordenada de destino, sem nenhuma maiúscula à frente. O número colocado à frente do lance, refere-se à ordem do lance dentro da partida. Finalmente, as capturas são representadas por um **x**, precedendo as coordenadas da casa de destino.

Por exemplo, o trecho **1.e4 d5 2.exd5 Dxd5** significa que, no primeiro lance do jogo, as Brancas moveram o seu peão da coluna **e** (peão-rei) até a casa **e4**, e as Negras moveram o seu peão da coluna **d** (peão-dama) até a casa **d5**. No segundo lance, as Brancas

capturaram o peão **d** negro com o seu peão **e**, e as Negras capturaram o peão branco com a sua dama.

Em uns pouco casos, duas peças de um mesmo tipo podem se dirigir ao mesmo destino, tornando necessário especificar a linha ou a coluna em que se situa a peça movimentada. No diagrama abaixo, os dois cavalos brancos, assim como os dois cavalos negros, podem se dirigir à casa **e5**. Então, as Brancas podem jogar **Cfe5** ou **Cde5**, enquanto as Negras podem jogar **C7e5** ou **C6e5** (apenas para efeito demonstrativo, porque todos esses lances, na posição apresentada, perdem uma peça).



Do lado direito de cada lance pode ser colocado um sinal que exprime a opinião do comentarista da partida:

! – lance muito bom; !! – lance extraordinário; ? – lance ruim; ?? – lance péssimo;
?! – lance duvidoso; !? – lance que merece atenção.

Vale observar que as palavras “Brancas” e “Negras”, no plural, e iniciando em maiúsculas, bem como as suas correspondentes em língua estrangeira, são usadas na bibliografia em substituição aos nomes dos jogadores, bem como para indicar o conjunto de todas as peças de uma determinada cor, o que possibilita abreviar e tornar mais elegantes os comentários das partidas.

Referências Bibliográficas

- AKL, Selim G. **The Design and Analysis of Parallel Algorithms**. Englewood Cliffs, New Jersey: Prentice Hall, 1989.
- ARKIN, Ronald C. **Behavior-Based Robotics**. Cambridge, Massachusetts: MIT Press, 1998.
- BAN, Amir. Entrevista concedida à Chessbase. Disponível em: <<http://www.chessbase.com/catalog/product.asp?pid=94>>. Acesso em: 02/05/2002.
- BONSDORFF, E.; FABEL, K.; RIIHIMAA, O. **Ajedrez y Matematicas**. Barcelona: Ediciones Martinez Roca, 1974.
- BRASSARD, Gilles; BRATLEY, Paul. **Fundamentals of Algorithmics**. Englewood Cliffs, New Jersey: Prentice Hall, 1996.
- BRONSTEIN, David. **El Ajedrez de Torneo (Zurich 1953 Candidatos)**. 2ª ed. Madri: Editorial Ricardo Aguilera, 1984.
- BUTTON, Graham. **Computadores, Mente e Conduta**. São Paulo: Fundação Editora da UNESP, 1998.
- CAMPBELL, Murray; HOANE Jr., Joseph A.; Hsu, Feng-hsiung. “Deep Blue”. In: **Artificial Intelligence**, 134(2002), p.57-83.
- CHESSBASE. **Advanced Chess – general info**. Disponível em: <<http://www.chessbase.com/events/events.asp?pid=133>>. Acesso em: 27/05/02.
- COLLINS, Harry M. **Experts Artificiels: Machines intelligentes e savoir social**. Paris: Éditions du Seuil, 1992.
- COLLINS, Harry M.; Kusch, Martin. **The Shape of Actions. What Humans and Machines Can Do**. Cambridge, Massachusetts: The MIT Press, 1998.
- DVORETSKY, Mark; YUSUPOV, Artur. **Positional Play**. London: B. T. Batsford Ltd, 1996.
- DREYFUS, Hubert L. **What Computers Still Can't Do: A Critique of Artificial Reason**. Cambridge, Massachusetts: The MIT Press, 1992.
- DUPUY, Jean Pierre. **Nas Origens das Ciências Cognitivas**. São Paulo: Editora da UNESP, 1996.
- FIDE. **Players & Ratings**. Disponível em: <<http://www.fide.org>>. Acesso em: 09/05/2002.
- FORD, Kenneth M., HAYES, Patrick J. On Computational Wings: Rethinking the Goals of Artificial Intelligence. In: **Scientific American Presents**, vol.9, no.4, 1998, p.78-83.
- GINSBERG, Matthew L. “Computers, Games and the Real World”. In: **Scientific American Presents**, vol.9, no.4, 1998, p.84-89.
- GRAU, Roberto G. **Tratado General de Ajedrez, Tomo Tercero: Estrategia – Conformaciones de Peones**. Buenos Aires: Editorial Sopena Argentina, 1958.
- _____. **Tratado General de Ajedrez, Tomo Quarto: Estrategia Superior**. Buenos Aires: Editorial Sopena Argentina, 1959.

- GRAUER, Ken; CAVALLARO, Daniel. **ACLS: Rapid Study Card Review**. 3ª ed. St. Louis, Missouri: Mosby Lifetime, 1997.
- HSU, Feng-hsiung et alli. "A Grandmaster Chess Machine". In: **Scientific American**, volume 263, no. 4, October 1990, pp.18-24.
- KROGIUS, N.V. **La Psicología en Ajedrez**. 3ª. edición. Barcelona: Ediciones Martinez Roca, 1974.
- KURZWEIL, Ray. **The Age of Spiritual Machines**. New York: Penguin Books, 2000.
- LAW, Averill M.; KELTON, W. David. **Simulation Modeling & Analysis**. 2nd ed. Singapore: McGraw Hill International Editions, 1991.
- LEVY, David; NEWBORN, Monty. **How Computers Play Chess**. New York: W.H. Freeman and Company, 1991.
- MCCARTHY, John. **AI as Sport**. Resenha de NEWBORN, Monty. Kasparov versus Deep Blue: Computer Chess Comes of Age. Disponível em: <www.formal.stanford.edu/jmc/newborn/newborn.html>. Acesso em: 30/01/2002.
- _____. **Making Computer Chess Scientific**. Disponível em: <www.formal.stanford.edu/jmc/chess.html>. Acesso em: 30/01/2002.
- PENROSE, Roger. **Shadows of the Mind: A Search for the Missing Science of Consciousness**. Oxford: Oxford University Press, 1994.
- POPPER, Karl. **O Conhecimento e o Problema Corpo-Mente**. Lisboa: Edições 70, 1997.
- PUCHKIN, V.N. **Heurística: A Ciência do Pensamento Criador**. 2ªed. Rio de Janeiro: Zahar Editores, 1976.
- REBEL. **GM Loek van Wely vs Rebel**. Disponível em: <<http://www.rebel.nl/reb-wely.htm>>. Acesso em: 05/05/2002.
- RICH, Elaine; KNIGHT, Kevin. **Artificial Intelligence**. 2ed. Singapore: McGraw Hill International, 1991.
- ROSZAK, Theodore. **O Culto da Informação: O Folclore dos Computadores e a Verdadeira Arte de Pensar**. São Paulo: Editora Brasiliense, 1988.
- RUSSELL, Stuart; NORVIG, Peter. **Artificial Intelligence: A Modern Approach**. Upper Saddle River, New Jersey: Prentice Hall, 1995.
- RYLE, Gilbert. **The Concept of Mind**. Chicago: University of Chicago Press, 1984 [1949].
- SEARLE, John. **The Rediscovery of Mind**. Massachusetts: MIT Press, 1992.
- _____. **Intencionalidade**. São Paulo: Martins Fontes, 1995.
- _____. **Mente, Cérebro e Ciência**. Lisboa: Edições 70, 1997.
- _____. "Minds, Brains and Programs". In: Cummins, Robert; Cummins, Denise D. (eds.). **Minds, Brains and Computers: The Foundations of Cognitive Sciences**. Malden, Massachusetts: Blackwell Publishers, 2000, p.140-152.
- SHANNON, Claude E. "Programming a Computer for Playing Chess". In: **Philosophical Magazine**, Ser.7, Vol. 41, No. 314 - March 1950.
- SHERESHEVSKY, M.I. **Endgame Strategy**. Oxford: Pergamon Press, 1985.

- TAIMANOV, Mark. Entrevista concedida à ChessBase. Disponível em:
<<http://www.chessbase.com/newsdetail.asp?newsid=320>>. Acesso em: 27/06/2002.
- TANENBAUM, Andrew. **Organização Estruturada de Computadores**. 3ª ed. Rio de Janeiro: Prentice Hall do Brasil, 1992.
- TEYSSANDIER, Jean-Paul ; COMBAULT, Jaques ; PECKER, Alain. « Rion-Antirion, le pont que défie les séismes ». In: **La Recherche**, nº 334, Septembre 2000, p. 42-46.
- UNIACKE, Mark. **Hiarcs 8**. Disponível em:
<<http://www.chessbase.com/catalog/product.asp?pid=112>>. Acesso em: 05/05/2002.
- WEIZENBAUM, Joseph. **Computer Power and Human Reason: From Judgment to Calculation**. San Francisco: W.H. Freeman and Company, 1976.
- ZNOSKO-BOROVSKY, Eugene. **The Middle Game in Chess**. London: G. Bell and Sons Ltd., 1930.