
**Hygor Piaget Monteiro Melo**

# Algoritmo genético para solução do problema da maioria

Fortaleza – CE

2011

**Hygor Piaget Monteiro Melo**

# *Algoritmo genético para solução do problema da maioria*

Dissertação de Mestrado apresentada ao Departamento de Física da Universidade Federal do Ceará, como parte dos requisitos para a obtenção do Título de Mestre em Física.

Orientador:
Prof. Dr. André Auto Moreira

Mestrado em Física
Departamento de Física
Centro de Ciências
Universidade Federal do Ceará

Fortaleza – CE

2011

Dissertação de Mestrado sob o título "Algoritmo Genético para Solução do Problema da Maioria", defendida por Hygor Piaget Monteiro Melo e aprovada em 11 de Agosto de 2011, em Fortaleza, Ceará, pela banca examinadora constituída pelos doutores:

---

Prof. Dr. André Auto Moreira
Departamento de Física – UFC
Orientador

---

Prof. Dr. José Soares de Andrade Jr.
Departamento de Física – UFC

---

Prof. Dr. Madras Viswanathan Gandhi Mohan
Departamento de Física – UFRN

# *Resumo*

Muitos sistemas naturais e sociais exibem comportamento globalmente organizado sem a presença de um controle central. Exemplos incluem convenções e normas, aprendizado social em animais e humanos, assim como modismos, boatos e revoltas. Exemplos em biologia também são abundantes: o comportamento evasivo de animais em grandes grupos, como peixes e pássaros, mostram uma grande sincronia mesmo na ausência de um líder. A fim de entender esses sistemas descentralizados, precisamos estudar primeiramente estratégias de coordenação global que utilizam apenas informações locais. Esse trabalho explora o uso do Algoritmo Genético na obtenção de estratégias naturalmente eficientes em ambientes ruidosos. O Algoritmo Genético é uma nova ferramenta importante na solução de problemas deste tipo, e oferece indícios de como a evolução deve atuar. Usando o que é conhecido sobre Algoritmos Genéticos, podemos descobrir mais sobre a evolução e seus mecanismos. A classificação por densidade é utilizada para testar o sucesso de estratégias, pois trata-se de um bom teste para coordenação global e processamento global de informações. Como é muito difícil evoluir regras com grande eficiência quando o número de vizinhos $k$ for maior que 5, isso sugere que a evolução deve construir soluções complexas baseadas em soluções de problemas simples. Usando essa ideia propomos um método de promover as regras aumentando o $k$. Com base na evolução inicial de regras com poucos vizinhos e usando o ruído como "pressão"evolutiva, nós fomos capazes de achar regras eficientes para um grande número de vizinhos, submetidas a condição de um alto nível de ruído. Achamos que as regras evoluídas são mais robustas a ambientes ruidosos do que a regra da maioria. A alta eficiência para grandes valores do ruído pode ser explicada em termos do maior peso dado por essas regras à informação da própria célula (não influenciada pelo ruído) do que a informação obtida através vizinhos. Como consequência, as células que empregam essas regras evoluídas tendem a manter seus próprios estados, até que uma grande maioria dos vizinhos discordem delas, mostrando um comportamento de persistência que pode ser encontrado em experimentos sociais.

# *Abstract*

Many natural and social systems exhibit globally organized behavior without the aid of a centralized control. Examples of such decentralized systems include conventions and norms, social learning in animals and humans, as well as fads, rumors and revolts. Examples are also abundant in biology: the evasive behavior of animals in large groups, such as fish and birds, show a great synchronicity even in the absence of an leader. In order to understand these decentralized systems, one must first understand strategies for global coordination that use only local information. This work explores the use of Genetic Algorithms in the creation of naturally efficient strategies in noisy environments. Genetic Algorithms are an important new tool in problem solving, and offer insight into how evolution may work. By using what is known about genetic algorithms, one can discover more about evolution and its mechanisms. The density classification task is used here to test strategy success, and revealed to be a good test for system-wide coordination and global information processing. Since it is very difficult to evolve highly fit rules when the number of neighbors $k$ is greater than 5, this suggests that evolution may build complex solutions based on solutions to simpler problems. Using this idea, we propose a method to promote rules increasing $k$. Based on the evolution of initial rules with few neighbors and using noise as evolutionary pressure, we were able to find efficient rules for a large number of neighbors, under the condition of a very high noise level. We find that the evolved rules are more robust to noisy environment than the majority rule. This increased efficiency at higher noise levels can be explained in terms of the larger weight given by these rules to the information of the evolving agent itself (not influenced by noise) than to the information obtained from its neighbors. As a consequence, the agents using these evolved rules tend to keep their own states, unless the great majority of their neighbors disagree with them, showing a persistence behavior that can be seen in social experiments.

# *Dedicatória*

*Dedico esse trabalho a minha namorada, família*
*e amigos por todo apoio dado...*

# *Agradecimentos*

Primeiramente gostaria de agradecer a todos os meus familiares, mas em especial aos meus pais Raimundo Rabelo Melo e Iracema Monteiro de Almeida Melo pela dedicação e apoio dados durante toda a minha vida.

Também muito importante para a realização deste trabalho foi meu orientador, o Professor Dr. André Auto Moreira junto ao Professor Dr. José Soares de Andrade Jr., aos quais agradeço pela paciência e compreensão que sempre tiveram durante o meu mestrado.

Durante todos os meu estudos sempre tive bons professores e sou muito grato a eles, em especial, aos professores Ascânio Dias Araújo, Fernando Antônio Amaral Pimentel, José Ramos Gonçalves, Josué Mendes Filho, Manuel Ferreira de Azevedo Filho, Marcos Antônio Araújo da Silva, Nilson Sena de Almeida, Ricardo Renan Landim de Carvalho, Eloneid Felipe Nobre, Antônio Gomes Souza Filho, Eduardo Bedê Barros, José Antônio Paixão, Constança Providência.

Também fiz muitos amigos, que me ajudaram sempre que possível: Diego Ximenes, Daniel Gomes, Daniel Marchesi, Davi Dantas, Diego Lucena, Diego Rabelo, Leandro Jader, Levi Leite, Rafael Alencar, Saulo Dantas, Vagner Bessa, Tiago Cerqueira, Márcio Ferreira, Carlos Perez, Heitor Credidio, Eduardo Araujo, Rilder Pires, Saulo Davi, César Menezes.

Não posso deixar de agradecer também a minha namorada Marianna Campos por toda a dedicação e compreensão durante esses 6 anos.

Agradeço, também, a todos os funcionários do departamento de Física da UFC e ao CNPq pelo apoio financeiro.

*"J'aimerais trouver les mots*
*Les mots justes, les mots qu'il faut*
*Mais tous les mots sont démodés*
*Tu sais*

*Alors j'écris je cherche encore*
*Le mot vrai*
*Le mot plus fort*
*J'ai l'impression qu'j'trouverais jamais*
*C'est vrai*
*Je sèche comme tu vois*
*Et toi"*
M – Ma Melodie

# *Sumário*

# *Lista de Figuras*

# *Lista de Tabelas*

# 1 Introdução

Muitos sistemas naturais e sociais exibem organização global de comportamento coletivo sem a ajuda de controle central. Exemplos incluem convenções e normas [**?**, **?**], aprendizado social em animais e humanos [**?**, **?**], assim como modismos, boatos e revoltas [**?**]. Exemplos em biologia também são abundantes: o comportamento evasivo de animais em grandes grupos, como peixes e pássaros, mostram uma grande sincronia mesmo na ausência de um líder. Em cada um desses exemplos, o indivíduo muda seu estado observando os estados dos seus vizinhos ligados por uma rede de comunicação. A emergência de um comportamento global nesses sistemas auto-organizados depende do desenvolvimento de estratégias baseadas em informação local. Essas estratégias não são necessariamente, ou pelo menos tipicamente, "racionais" ou "livres de erro". Em vez disso, os indivíduos podem cometer erros, podem ser propensos a erros de processamento, e podem precisar confiar em informações incompletas, possivelmente corrompidas. Surpreendentemente, estratégias simples apresentam boa eficiência em muitos ambientes experimentais [**?**].

Uma fonte de complexidade em sistemas reais é a estrutura da rede de comunicações. A típica idealização feita de indivíduos localizados em uma rede regular interagindo com seus vizinhos mais próximos dessa matrix é claramente inadequada. Recentemente estudos tem demonstrado que as estruturas das redes de interação de sistemas biológicos, sociais e tecnológicos definitivamente não são triviais [**?**, **?**, **?**]. A topologia complexa é conhecida por assim afetar os processos dinâmicos e pode ter uma forte influência na emergência do comportamento coletivo [**?**].

Uma importante característica dos sistemas complexos é o fato de se adaptarem ou evoluírem de acordo com o "ambiente" interno e externo ao sistema, tornando ele mais robusto a mudanças. É importante saber como esse processo de evolução e adaptação ocorre. O processo de Evolução natural pode ser utilizado como método de otimização e resolução de problemas, através da utilização do Algoritmo Genético. Podemos, também, utilizar o Algoritmo Genético na solução de um problema de modelagem, buscando entender como dá-se o aparecimento de modo "natural" das estratégias de coordenação global

e como essas estratégias se adaptam a uma topologia complexa, e a defeitos ou ruídos na informação vinda de seus vizinhos.

Nesse trabalho exploramos o uso do Algoritmo Genético na obtenção de estratégias naturalmente eficientes em um ambiente ruidoso e há uma topologia complexa. O Algoritmo Genético é uma importante nova ferramenta na solução de problemas, e oferece indícios de como a evolução natural deve ter ocorrido. Adotamos como tarefa computacional o problema da maioria pois é um teste eficiente para estratégias de sucesso em coordenação global e processamento de informação. Construímos um método capaz de usar o Algoritmo Genético na busca de soluções mais complexas para o problema da maioria. A eficiência do Algoritmo Genético na busca de regras com um número pequeno de vizinhos já havia sido revelada em por outros trabalhos [**?**, **?**]. A evolução natural deve construir soluções para problemas complexos utilizando como padrão a solução para problemas mais simples, inspirados nisso, propomos um método de promoção que permite que através do Algoritmo Genético achemos a solução para um número grande de vizinhos. Partindo da evolução inicial de regras para poucos vizinhos e usando o ruído como pressão evolutiva, fomos capazes de encontrar regras eficientes com um número grande de vizinhos.

Finalmente buscamos realizar um estudo detalhado dessas regras evoluídas com o objetivo de justificar a eficiência delas. A robustez, definida como a dependência da eficiência da regra pelo ruído existente, foi calculada e comparada com uma regra heurística conhecida como regra da maioria. O comportamento inesperado da robustez quando comparado com a regra da maioria nos levou a um estudo detalhado de como as regras evoluídas resolviam o problema da maioria, e como podiam ser mais robustas que a própria regra da maioria. Esse estudo demonstrou que um comportamento conservador naturalmente evolui como modo de defesa para um ambiente com grande ruído, mostrando-se eficaz na solução do problema da maioria.

# 2 Complexidade

A área de sistemas complexos vem sendo tema, nos últimos 20 anos, de muitas publicações científicas e tecnológicas. Apesar disso não há ainda uma definição formal de sistema complexo amplamente aceita, parte dessa ambiguidade é devido ao fato do comportamento complexo emergir em disciplinas diversas, tornando-a uma área extremamente interdisciplinar. Essa interdisciplinaridade, apesar de trazer dificuldades formais, é também onde está o grande potencial do estudo de sistemas complexos, pois esse estudo promoveu ligações entre áreas como física, biologia, química, economia, sociologia, linguística, neurociência, dentre outras.

Informalmente, podemos considerar que um sistema complexo é uma grande rede com constituintes relativamente simples, que interagem entre si, sem um controle central, e que, mesmo assim, apresenta um comportamento complexo emergente [**?**]. Devemos esclarecer duas coisas nessa definição: O que seria relativamente simples e o que seria um comportamento complexo emergente. O "relativamente simples" se refere a uma comparação entre o comportamento individual do componente ao comportamento da estrutura coletiva. Por exemplo, no caso de uma formiga, é claro que ela como animal apresenta um alto grau de complexidade, mas o seu comportamento individual pode ser considerado simples quando comparado com o comportamento da sua colônia. Isso está diretamente ligado à não-linearidade desses sistemas. A grosso modo, o comportamento do todo não pode ser entendido como a soma do comportamento das partes.

Já para o caso do *comportamento complexo emergente*, ele se refere ao fato de que o comportamento global do sistema não é somente complexo, mas surge da ação de componentes simples cujo mapeamento individual no comportamento global é não-trivial, como demonstra a Fig. 2.1.

De certo modo, não podemos considerar o que acabamos de citar como uma definição rigorosa, pois a todo momento utilizamos a palavra "complexo", que obviamente não definimos. De qualquer forma as propriedades que citamos podem nos dar uma boa idéia

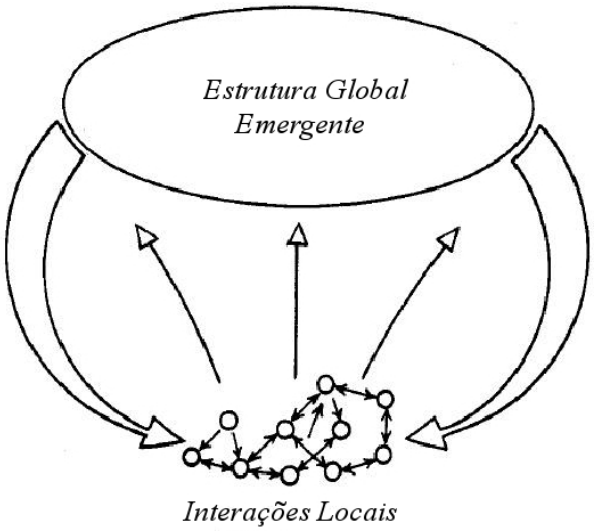



Figura 2.1: Complexidade Emergente. A estrutura global emergente do sistema é resultado de simples interações locais entre vários constituintes, mas cujo mapeamento nessa estrutura é intrincado. Ou seja, não podemos tomar um comportamento global específico e tentar explicá-lo baseado apenas em função da forma da interação local. Ele só pode ser entendido quando levamos em conta todas as interações imersas em uma rede de topologia não-trivial (retirado de [**?**]).

do que seja um sistema complexo, nos permitindo identificá-lo quando estivermos diante dele.

Podemos citar várias características que são recorrentes em sistemas classificados como complexos [**?**], mas iremos nos focar apenas em duas por serem as principais características do sistema a ser estudado durante esse trabalho. A primeira é o fato do sistema ser um *Sistema Descentralizado* e a segunda é o fato dele ser *Adaptativo*.

## *Sistema Descentralizado*

A maioria dos sistemas complexos são constituidos de uma rede grande de componentes, em que cada um segue um regra relativamente simples, em que atualiza o seu estado baseado nos estados de um número pequeno de vizinhos, quando comparado com o tamanho da rede, ou seja, não há a presença de um controle central ou "líder". A presença de um controle central tornaria a maioria das tarefas triviais. Um exemplo é o *Problema da Maioria* [**?**] em que temos um sistema complexo, com um conjunto grande de "células" podendo ter dois estados acessíveis. Inicialmente temos uma distribuição de estados entre essas células, com um dos estados sendo mais recorrente. Com o passar do tempo, as células devem atualizar seus estados de acordo com os seus vizinhos para que no fim todas tenham atingido o consenso, ou seja, todas encontrem-se no mesmo estado, o estado mais

comum no início. Nesse caso, se tivéssemos um controle central, uma dada célula poderia apenas contar todos os estados e verificar qual o de maioria e depois "ordenar" que todas as outras mudassem seu estado para o estado de consenso.

O Comportamento Coletivo Complexo muitas vezes está ligado à topologia complexa da rede de interação dos componentes do sistema. Desta forma, o grande avanço atual na teoria de redes complexas, tornou possível a análise e modelagem de sistemas complexos baseados nas propriedades dessas redes [**?**].

## *Adaptação*

Uma propriedade importante de sistemas complexos que será bastante explorada no nosso trabalho é o fato de se adaptarem ou evoluírem de acordo com o "ambiente" interno e externo ao sistema, tornando este robusto a mudanças, característica bastante procurada por ter grande importância prática. O processo de *Evolução Natural*, por sua vez, pode ser também utilizado como método de otimização e resolução de problemas computacionais, através de um tipo de algoritmo chamado de *Algoritmo Genético*, que será utilizado no nosso trabalho e por esse motivo terá um capítulo dedicado à sua explicação detalhada. Sendo assim, essas características tornam esse ramo de estudo bastante rico e vasto, podendo ir desde o estudo da origem das espécies até o estudo de crises econômicas. Podemos ver agora com mais detalhes alguns exemplos clássicos de sistemas com comportamento complexo:

1. ***Colônia de Insetos***

   Colônias de insetos sociais são exemplos ricos de sistemas complexos de fácil visualização com os quais convivemos diariamente. Há muitas espécies que apresentam essa propriedade, dentre elas as colônias de formigas, abelhas e cupins, são talvez os exemplos mais comuns a serem lembrados. No caso das formigas, podemos ter até comunidades de centenas de milhares de indivíduos, cada um obedecendo seus instintos genéticos, para buscar comida, seguir trilhas de sinais químicos de outras formigas da colônia, enfrentar intrusos, entre outras atividades. No entanto, como podemos observar, todos os indivíduos realizando suas tarefas simples compõem uma única colônia de comportamento emergente complexo, essencial para a sobrevivência da colônia como um todo. É importante notar que não há um líder dando ordens a cada formiga, do contrário as tarefas realizadas seriam triviais. No caso, como já explicamos anteriormente, uma das propriedades que tornam um sistema complexo é o fato dele realizar tarefas complexas, onde apenas regras individuais

simples são consideradas.

Figura 2.2: Fotografia que mostra um grupo de formigas construindo uma ponte com seus próprios corpos para permitir que a colônia tome o caminho mais curto através de uma fenda [**?**].

Desse modo, há muitas perguntas a serem respondidas. Como indivíduos guiados apenas por suas vontades, moduladas apenas pela opinião do seus vizinhos mais próximos, podem produzir enormes estruturas robustas como coméias, ou complexas redes de tubos subterrâneos, de modo a promover o bem e a sobrevivência da colônia, muitas vezes em detrimento de sua própria sobrevivência? Como tais comunidades podem se utilizar de estratégias indiretas para obter alimento, como no caso de muitas espécies de formigas, onde todo alimento capturado por elas é levado para o formigueiro não para se alimentarem diretamente dele, mas para servir de cultivo de um fungo do qual elas se alimentam? Como tudo isso pode ter aparecido através da evolução? Esse tipo de organização tornou as formigas, e outros insetos sociais, uma das mais eficientes espécies. As formigas constituem uma única família, *Formicidae*, com mais de 12.500 espécies classificadas distribuídas por todas as regiões do planeta, exceto nas regiões polares. As formigas são o gênero animal de maior sucesso na história terrestre, constituindo de 15% a 20% de toda a biomassa animal terrestre [**?**].

2. ***O Cérebro***

Sem dúvida o cérebro pode ser considerado como um sistema extremamente complexo. A nossa compreensão pobre dele ainda é superficial, mas vem avançando

cada vez mais, junto ao avanço no estudo de sistemas complexos. Seu estudo é essencial e talvez a sua compreensão seja um dos principais objetivos da ciência. Novamente, o sistema aqui mensionado é uma grande rede constituída de pequenas células chamadas neurônios. Na verdade ele é também constituído de outros tipos de células, mas a maioria dos neurocientistas acreditam que o pensamento, sentimento, percepção, consciência e todas as outras atividades de larga escala são produzidas pelos neurônios e pelo padrão de ligação do seus grupos [**?**].

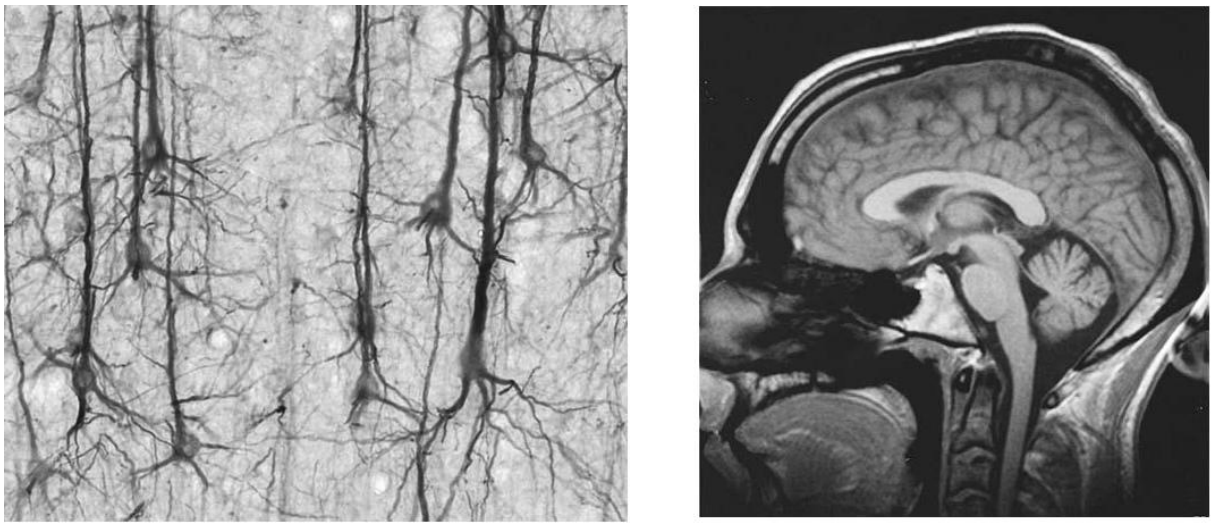


Figura 2.3: *Esquerda.* Visão microscópica dos neurônios formados pelos *dendritos*, *axônios* e *soma*. *Direita.* Cérebro humano que possui todas as suas ações macroscópicas complexas governadas microscopicamente pelos neurônios [**?**].

Como visto na Fig. 2.3, os neurônios são formados basicamente pelos *neuritos* (*dendritos* e *axônios*) e pelo *soma*. Cada neurônio está ligado em média a $10^4$ outros neurônios. Tendo em vista que temos cerca de $10^{11}$ neurônios, essa associação entre eles constitui o que chamamos anteriormente de comportamento coletivo complexo. Com efeito, o cérebro como um todo processa e guarda informações baseado no comportamento individual dos neurônios, que por sua vez tem apenas acesso direto à informação limitada pelos seus vizinhos mais próximos. A comunicação dos neurônios se dá através de sinais elétricos propagando-se pelos dendritos em direção ao soma que, dependendo da intensidade do total de sinais elétricos vindos de todos os dendritos, emite um novo pulso elétrico, o qual viaja pelo axônio em direção às suas extremidades. Sendo assim, as capacidades de pensar, sentir, decidir, lembrar, etc. podem ser consideradas como propriedades emergentes de redes neurais compostas por um número muito grande de neurônios interagindo de modo não linear, como

um sistema complexo [**?**]. Ainda não se entende como as ações individuais junto à rede formada pelos neurônios podem gerar o comportamento global do cérebro. Não se sabe qual é o número exato de neurônios que trabalham juntos para produzir o comportamento cognitivo, ou como exatamente isso pode gerar nossos pensamentos e o processo de aprendizado. Ainda não se conhece também, como todo esse sistema e habilidades podem ter aparecido através da evolução natural.

3. ***Economias***

A economia é outro exemplo de sistema complexo, em que os componentes são as pessoas, ou companhias, relacionadas através de venda e compra de produtos, ou serviços, e que como todos os outros exemplos apresenta um comportamento global complexo. Para isso basta imaginar o quão difícil é a previsão de eventos no mercado financeiro. Outro fator importante é o fato da Economia ser um sistema adaptativo tanto na micro como na macroescala. Em escala micro, indivíduos e companhias tentam melhorar sua rentabilidade observando e aprendendo com o comportamento de outros indivíduos e companhias. Pensava-se classicamente que os indivíduos tomavam sempre decisões racionais de modo a otimizar seu retorno fazendo com que a livre competição produzisse um estado de equilíbrio estável, estado esse em que a tecnologia mais eficiente, no momento, dominaria o mercado. A estabilidade do equilíbrio supostamente seria garantida pelo fato das leis que controlam os valores de mercado serem leis de *feedback negativo*.

Nesse ponto é importante uma pequena digressão para entendermos o que seria uma lei de *feedback negativo*. Podemos tomar como exemplo uma válvula utilizada antigamente em motores a vapor. A válvula tem como função mater constante a pressão do gás que sai da caldeira. Para isso quando o motor começa a girar mais rápido, consequência do aumento da vazão de gás, a válvula se fecha um pouco, o que diminui a vazão de gás e com isso diminui também a velocidade do motor. Ao diminuir a velocidade do motor, a válvula responde aumentando a abertura, o que faz com que a vazão aumente, retornando ao início. Fica claro que esse tipo de resposta que a válvula tem ao aumentarmos ou diminuirmos a velocidade do motor faz com que atinjamos um estado estável de vazão constante. Caso tivéssemos um sistema de *feedback positivo*, a válvula deveria aumentar a vazão de gás, caso a velocidade do motor aumentasse. Isso teria como consequência o colapso do motor, ou a sua parada, pois caso eventualmente haja um aumento da velocidade do motor, a válvula ira abrir ocasionando um aumento maior da velocidade do motor e assim sucessivamente, até que o motor entre em colapso. O conceito de *feedback negativo e*

*positivo* é também importante em biologia, pois aparece no contexto de uma possível explicação para o pavão macho ter calda tão grande, pois se trataria obviamente de uma desvantagem na fuga de predadores. De modo geral, o processo de seleção natural é de *feedback negativo*, mas a explicação mais aceita para os casos extremos como o dos pavões é de que a seleção das fêmeas, nesse caso, é uma lei de *feedback positivo* incentivando o crescimento exagerado da calda, ou seja, as fêmeas gostavam de caldas grandes e tinham filhos com machos de caldas grandes gerando novas fêmeas que gostam de caldas cada vez maiores. É obvio que esse processo tem um limite, pois chegará um ponto onde o pavão macho não conseguirá mais sustentar a sua calda. Talvez antes mesmo ele será morto por um predador, mas mesmo assim a espécie levará a uma situação extrema onde claramente não há vantagem de se ter uma calda tão grande [**?**].

Voltando à Economia, essa visão clássica não explicaria as crises econômicas e as bolhas especulativas, fenômenos típicos de um sistema de resposta positiva. Um simples exemplo disso está no famoso dilema econômico conhecido como *Tragédia dos comuns* que foi popularizado por Garrett Hardin quando publicado em 1968 na revista científica *Science* [**?**]. Nesse trabalho, Hardin mostra que o livre acesso e a demanda irrestrita de um recurso finito acaba por condená-lo, devido a uma superexploração. Para isso ele constrói um cenário simples de uma pastagem compartilhada por pastores locais. Assumindo que os pastores desejam maximizar a sua produção, então a cada ovelha inserida temos um efeito positivo e negativo. Positivo, pois o pastor receberá o lucro sobre esse novo animal, e negativo, pois a pastagem é cada vez mais degradada com os novos animais adicionados. O ponto principal é que a divisão dos lucros e prejuízos é desigual, considerando que os lucros vão todos para o dono das ovelhas enquanto o prejuízo é dividido entre todos os pastores. Caso todos os pastores pensem dessa forma, isso levará a uma super exploração da pastagem, pois todos desejarão inserir novas ovelhas, aumentando seu lucro. Outro exemplo prático, é a exploração do mar pela pesca. Como "todos" podem pescar, cada companhia de pesca busca aumentar seus lucros melhorando seus barcos e assim distribuindo o prejuízo da falta de peixes para todos, levando a situações críticas de diminuição na população de certas espécies de peixes superexploradas.

4. ***A "World Wide Web"***

Desde o seu nascimento no CERN no início dos anos 90, a WWW ("World Wide Web") vem crescendo exponencialmente. Assim como nos outros exemplos, a Internet pode ser pensada como um sistema complexo auto-organizado, no sentido de se

tratar de “indivíduos” conectados em uma topologia complexa e realizando tarefas simples, como postar novas páginas na ”Web”, criando novas conexões através de links nessa página.

Quando estudada como uma rede complexa, a WWW apresenta propriedades estatísticas presentes em muitos outros sistemas complexos. O que os pesquisadores descobriram é que a estrutura global da rede WWW apresenta uma propriedade estatística conhecida como livre de escala [**?**]. Como cada sítio está conectado com um número $k$ de outros sítios, isso significa que se fizermos um gráfico da frequência do número de links, $P(k)$, para todos os valores de $k$, veremos que essa frequência escala com uma potência de $k$, onde $\gamma$ é o expoente da lei de potência (ver Fig 5). Matematicamente temos $P(k) \propto k^{-\gamma}$. Portanto, o número de sítios com $k$ sítios a ele conectado, através de links, decresce segundo uma potência de $k$, com $\gamma = 2.1$ [**?**]. O interessante é que esse tipo de distribuição é bastante recorrente em outros sistemas complexos, como a famosa *lei de Gutenberg e Richter*: A frequência anual de terremotos em uma dada região escala lineramente em logaritmo com a sua magnitude, portanto terremotos de grande magnitude são bem menos frequentes que terremotos de baixa magnitude [**?**]. No caso da WWW, essa propriedade está diretamente ligada ao modo como a rede cresceu e vem crescendo. Tal crescimento está intimamente relacionado com sua função de propagar informação como um novo meio de comunicação. É interessante notar que essa propriedade estatística está associada ao com o grande sucesso do sistema de busca mais utilizado atualmente, o *Google*. Esse sistema de busca indexa seus sítios de acordo com o número de links que cada sítio recebe na Internet, ou seja, o *Google* parte do princípio de que quanto mais links um sítio recebe mais importante ele seria. Desse modo, como a frequência de sítios com números grandes de links cai rapidamente com a lei de potência mostrada anteriormente, isso faz com que geralmente a busca seja eficiente, já para os primeiros sítios propostos como resultados. Por esse motivo, é comum não se passar da primeira página para se achar o resultado desejado pela busca.

De certo, a WWW se mostrou um excelente ambiente de testes, em que se pode realizar diversos experimentos com o intuito de se medir e observar a validade dos modelos produzidos pelo estudo de redes complexas. Sem o seu aparecimento, provavelmente não teríamos tido muitos avanços nessa área.

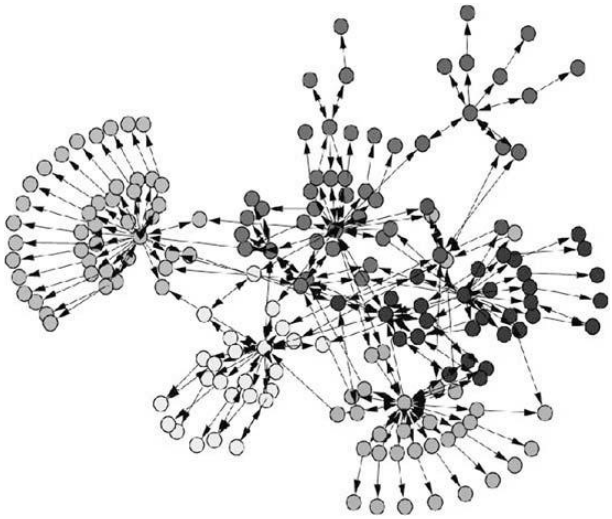


Figura 2.4: Representação da estrutura de uma pequena parte da WWW. Os sítios representam as páginas na rede e as ligações os *hiperlinks* entre as páginas (retirado de [**?**]).

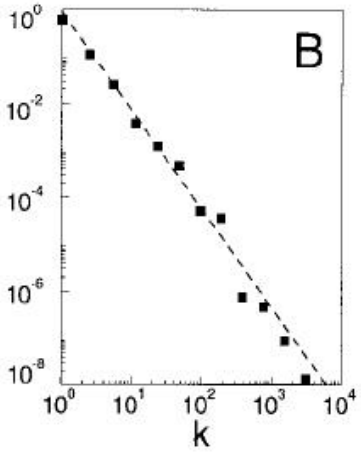



Figura 2.5: Distribuição de conectividade da rede WWW, mostrando o comportamento em lei de potência encontrado [**?**].

Vemos que a lista de exemplos acima é pequena, pois o número de sistemas naturais com comportamento complexo é enorme. O completo entendimento das propriedades desses sistemas constitui um dos maiores desafios da ciência atual. Assim como a Teoria Quântica criou uma ponte entre a Física e a Química, a compreensão da complexidade tem como promessa ligar a Física à Biologia, permitindo a modelação e compreensão de sistemas vivos. Portanto, o estudo de sistemas complexos é de extrema importância, pois tem como promessa trazer um grande avanço em diversas áreas da ciência, desde a Física até a Biologia.

# 3 Algoritmo Genético

Nos anos 50 e 60 muitos cientistas computacionais e engenheiros, de modo independente, trabalharam estudando sistemas evolutivos com a idéia de que o princípio da evolução poderia ser usado como uma ferramenta de otimização para problemas de engenharia e física [?]. No início de 1948, Turing propôs uma "busca genética ou evolutiva", e em 1962 Bremermann executou experimentos computacionais em uma "otimização através de evolução e recombinação" [?]. Durante os anos 60, apareceram três diferentes implementações do que chamamos hoje de Computação Evolutiva. Nos EUA, Fogel, Owens, e Walsh introduziram a *programação evolutiva* [?], técnica na qual as soluções candidatas são representadas como máquinas de estados finitos, e são evoluídas mutando-se aleatoriamente seus diagramas de transição de estado e selecionando-se as mais aptas. Já na Alemanha, Rechenberg e Schwefel inventaram o algoritmo de *estratégias evolutivas* [?, ?], que teve como uma das primeiras aplicações a otimização de parâmetros usados na construção de dispositivos como aerofólios. Enquanto isso, Holland desenvolvia o método chamado *Algoritmo Genético* [?, ?]. Muitos outros trabalharam e desenvolveram outros algoritmos inspirados na evolução para otimização ou *aprendizagem de máquinas* (*learning machines*) durante os anos 50 e 60, Box [?], Friedman [?], Bledsoe [?], Bremermann [?], e Reed, Toombs e Baricelli [?], porém o maior impacto foi dado pelo Algoritmo Genético, a Programação Evolutiva e a Estratégia Evolutiva que constituem hoje a base do que se chama Computação Evolutiva. A Tabela 3.1 mostra uma analogia simples entre a Evolução natural e um algoritmo de otimização.

| Evolução | Otimização |
|---|---|
| Ambiente | Problema |
| Indivíduo | Solução Candidata |
| *Fitness* | Qualidade |

Tabela 3.1: Analogia entre a Evolução natural e um esquema básico de computação evolutiva aplicado a um problema de otimização.

Como citamos antes, o Algoritmo Genético (GA, *Genetic Algorithm*) foi inventado no anos 1960 por John Holland, e foi continuamente desenvolvido por Holland e seus estudantes e colegas nos anos 60 e 70, na Universidade de Michigan. Diferente dos outros algoritmos evolutivos, que foram desenhados com o objetivo de resolver problemas de otimização principalmente em engenharia, o GA foi pensado por Holland para ser um meio de se estudar formalmente os fenômenos de evolução e adaptação da natureza através de simulações computacionais.

O Algoritmo Genético é um modelo que permite partir de uma população de "cromossomos" (como exemplo podemos imaginar vetores com valores 1 ou 0, ou *bits*) para atingir uma nova população usando um tipo de "seleção natural", utilizando operadores inspirados em genética: *crossover*, mutação, inversão. Cada cromossomo consiste de uma compilação de "genes", em que cada gene tem um determinado "alelo" (e.g. 0 ou 1). O operador seleção, semelhante à natureza, realiza uma escolha, ou classificação, dos cromossomos que irão reproduzir para gerar uma nova população. Essa escolha, ou classificação, é feita através de uma função de *fitness*, que basicamente mede a qualidade daquela solução, ou quão próxima ela está da solução ótima. O crossover é um operador que troca partes de dois cromossomos ("pais") gerando um novo cromossomo ("filho"), mimetizando a recombinação biológica entre dois organismos haploides. A mutação ocorre escolhendo-se um gene aleatoriamente e modificando seu alelo também de modo aleatório.

A inovação de Holland foi a introdução de um algoritmo baseado em uma população com os operadores de crossover e mutação. A Estratégia de Evolução de Rechenberg começava apenas com dois indivíduos, um pai e um filho, com o filho sendo uma mutação do pai. Do mesmo modo, na Programação Evolutiva, o único fator que introduz diversidade é a mutação. Holland também foi o primeiro a produzir uma base teórica para computação evolutiva [**?**].

## 3.1 Evolução Natural

A publicação por Darwin do livro "Origem das Espécies" [**?**] hoje é visto como uma das principais conquistas da ciência. Nele se encontra um modelo que explica a diversidade biológica e sua complexidade através de primeiros princípios. De modo mais específico, Darwin propôs uma teoria da evolução de um ponto de vista macroscópico [**?**]. De fato, na sua época a teoria genética ainda não era conhecida. Para Darwin, a seleção natural tinha um papel central na sua teoria da evolução. A seleção natural ocorre através da

competição por recursos, modulados pelo ambiente e pela reprodução. Sendo assim, através do mecanismo da reprodução, os indivíduos mais bem "sucedidos" na utilização dos recursos, podem passar seu material genético através das gerações. Esse processo de seleção naturalmente promove uma melhora, ou aumento, da eficiência na sobrevivência de um grupo de indivíduos, mas isso não explicaria o aparecimento de novas "características", pois como nunca há uma população infinita de indivíduos, o número de genótipos diferentes é limitado. Portanto, o processo de adaptação, relacionado com a robustez desse tipo de sistema, deve prever o aparecimento de novas características, do contrário isso limitaria a abrangência da capacidade da seleção. Darwin identificou que era necessário uma variação aleatória nas características de uma dada população.

Os traços fenotípicos são as habilidades físicas e comportamentais de um indivíduo que diretamente afetam a sua resposta ao ambiente determinando o seu *fitness*. Cada indivíduo é uma combinação de fenótipos (microscopicamente relacionado aos genes) que são avaliados pelo ambiente. Dependendo dessa avaliação, esse o conjunto de genes relacionados propagam-se no tempo através da reprodução, momento em que há uma combinação do material genético entre dois indivíduos em uma população. Desse modo, como citado anteriormente, a descoberta de Darwin foi que pequenas variações aleatórias em um dado genótipo (mutações) devem ocorrer no processo de reprodução. Essas novas características, geradas pela mudança do material genético, serão então testadas através de gerações por meio da seleção.

Esse processo pode ser visualizado através da teoria da paisagem (landscape) de *fitness* criada por Sewall Wright [**?**], como visualizada na Fig. 3.1. Nela vemos duas dimensões relacionadas a dois tipos diferentes de fenótipos (e.g. altura e peso), e para cada fenótipo temos uma escala de valores possíveis. A terceira dimensão é a medida do *fitness*, ou seja, para cada conjunto de valores de fenótipos temos uma "altura" de *fitness*. A população é representada nesse mapa como um conjunto de pontos sobre essa superfície, onde cada ponto representa um indivíduo com sua particular combinação de fenótipos.

Um exemplo prático da utilidade dessa abordagem pode ser encontrado no trabalho de Butch Brodie [**?**]. Brodie produz um landscape correlacionando o comportamento diante de um predador da cobra *Thamnophis ordinoides* e o padrão da sua coloração, como mostra a Fig. 3.2. De fato, observa-se que essas cobras têm uma vasta variação no padrão da sua coloração, indo desde o padrão malhado até o de listras longitudinais. Essa característica foi medida por Brodie, e é representada no gráfico pela medida de *stripedness*. As cobras também apresentam uma variação segundo seu comportamento

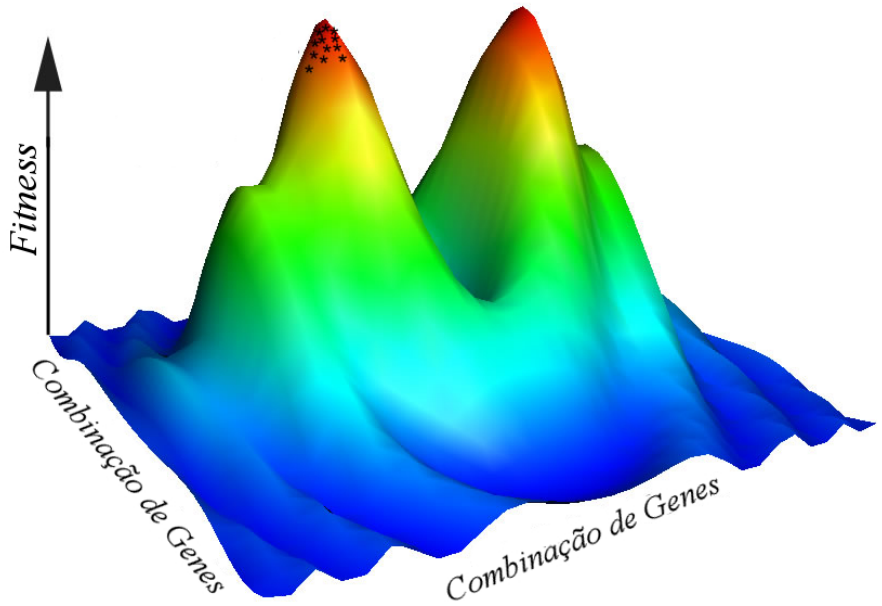



Figura 3.1: Landscape adaptativo. Visão criada por Wright ao propor que podemos imaginar a evolução como uma subida de uma "montanha", em que a altura significaria o *fitness* e as dimensões do plano seriam as características fenotípicas ou genéticas. No gráfico temos que o eixo $z$ representa o *fitness* enquanto que $x$ e $y$ são os valores de dois genes diferentes.

diante de um potencial predador. Algumas movem-se ao longo de uma pequena distância e repentinamente revertem sua direção, enquanto outras preferem fugir em um movimento aproximadamente linear. Isso está relacionado com o fato de que as cobras listradas são mais fáceis de se vizualizar paradas do que quando estão se movendo linearmente, pois as suas listras longitudinais dificultam a determinação visual da sua velocidade. Para as cobras malhadas tem-se o contrário, posto que quando paradas, tendem a se camuflar com o solo, tornando mais difícil capturá-las do que quando se movem linearmente. A combinação dessas características produz dois pontos com alto *fitness* e outros com *fitness* mais baixo, como mostrado na Fig. 3.2.

A evolução pode ser entendia como o processo gradual, causado pela seleção natural e pelas mutações, de subida de uma dada população aos picos presentes em uma paisagem [**?**]. Esse problema pode ser dividido em duas classes. Caso dentro de todo o domínio das soluções haja apenas um pico onde o valor do *fitness* é maior do que em todos os seus vizinhos, podemos chamar esse problema de *unimodal*. No outro caso, temos o problema *multimodal*, em que no espaço de soluções podemos encontrar mais que um pico de *fitness*. Os picos também podem ser classificados de dois modos, podem ser máximos locais ou globais. Um máximo local é todo e qualquer máximo, ou seja, basta que ele seja um ponto cujo *fitness* é superior a todos os seus vizinhos. O máximo global é o maior de

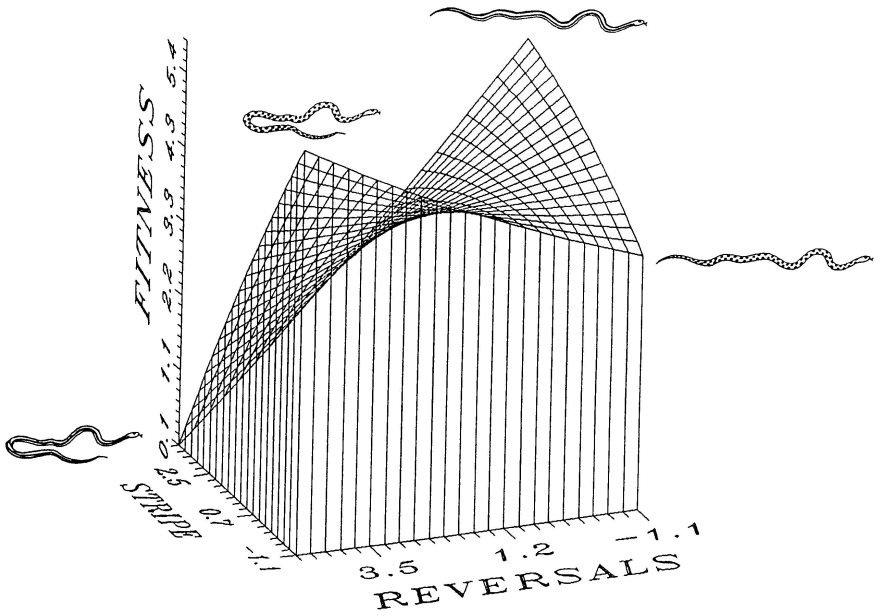



Figura 3.2: Landscape adaptativo para as cobras *Thamnophis ordinoides*. O gráfico mostra a dependência do *fitness* com o comportamento evasivo e a variação de coloração dessa espécie de cobra. O eixo nomeado *Reversal* mede o quão tortuosa é a trajetória da cobra na evasão sobre um suposto predador, ou seja, o quanto a trajetória é diferente de uma reta. A outra característica medida é a indicada no gráfico como *Stripe*, que mede quanto a coloração das cobras são de listras longitudinais, ou seja, quanto menor esse valor, mais "malhada" é o padrão de cor de pele da cobra. Podemos ver que há dois picos, quando a cobra é "malhada" ao máximo ela deve ter a sua trajetória de fuga o menos linear possível para ser eficiente. Já o outro pico corresponde à cobra que possui o máximo de padrão de listras e o mínimo de *reversal*, ou seja, àquela que foge segundo uma linha reta (figura retirada do trabalho [**?**]).

todos os locais. Ou seja, um máximo global também é um máximo local, enquanto nem todo máximo local é global.

Observando a Fig. 3.3, podemos identificar o mesmo problema notado por Wright: "o problema da evolução é o mecanismo na qual as espécies podem continuamente achar o caminho entre os pequenos picos e os mais altos" [**?**]. Como podemos ver, o processo clássico de evolução nos levaria a um problema, de fato a população evoluiria de modo a ficar presa em máximos locais, não sendo permitido atravessar esse vale para atingir um pico mais alto. Wright identificou esse problema e tentou resolvê-lo através da teoria do *Shifting Balance* [**?**]. Nessa teoria, o fator importante para sair dos máximos locais é a diversidade genética da população relacionada ao conceito de *drift genético*. Pela teoria de Wright o máximo de um *landscape* qualquer poderia sempre ser alcançado.

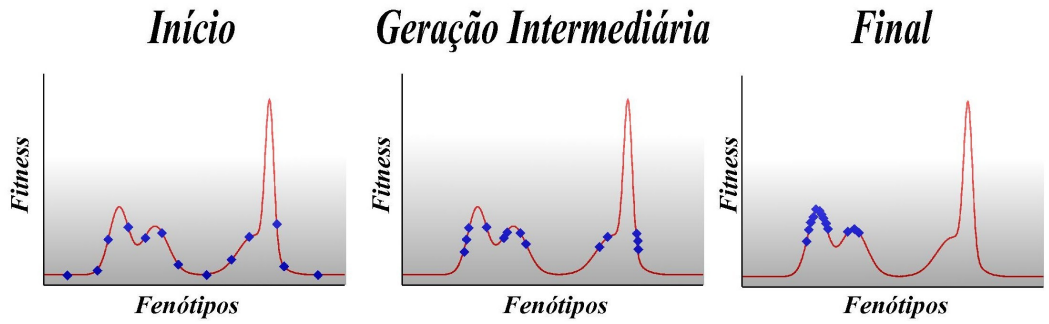



Figura 3.3: Evolução sobre um landscape adaptativo. A evolução pode ser vista como um processo de subida sobre uma superfície em que temos uma dimensão associada ao *fitness* e todas as outras representando os traços genéticos ou fenotípicos. Há uma tendência gerada pela seleção natural levando a favorecer sempre animais que tenham a combinação de características com o maior *fitness*. O que podemos ver é que a medida que as gerações vão passando, a população se acomoda em máximos locais, tornando difícil a tarefa de se achar um máximo global.

## 3.2 Genética

Do ponto de vista microscópico, a teoria de evolução é explicada pela genética molecular. Mesmo antes da ciência, os humanos já tinham um conhecimento rudimentar sobre genética, mais precisamente sobre hereditariedade, pois realizavam domesticação e cruzamentos seletivos de animais e plantas. Até que em 1866, Gregor Mendel realizou uma série de experimentos em que pôde estabelecer os padrões de hereditariedade de algumas características da planta da ervilha. Seu triunfo deu-se, dentre vários outros motivos, por ele ter abordado o problema sobre um ponto de vista mais rigoroso utilizando-se de análise estatística. Todavia, é interessante que seu trabalho só foi ser realmente descoberto pela comunidade científica no início do século XX, quando Mendel já havia falecido.

A hipótese fundamental da genética é que cada indivíduo tem uma estrutura dual: suas propriedades fenotípicas, e a representação dessas propriedades contidas em um nível menor, pelo seu *genótipo*. Ainda há muita discussão sobre esse assunto, mas hoje se pensa que os traços fenotípicos são uma combinação entre o genótipo e o ambiente, contudo podemos imaginar que o genótipo "computa" o fenótipo.

O material genético de um organismo está contido em alguns cromossomos. Em seres "superiores" (e.g., animais e plantas) os cromossomos se organizam em pares na maioria das células, sendo assim chamados seres *diploides*. Em uma célula humana, exceto a dos gametas, temos 23 pares de cromossomos. Já as células que contenham apenas um complemento dos cromossomos são chamadas de *haploides*. Desse modo, a combinação

do material genético paternal e maternal em um organismo diploide ocorre pela fusão do material dos dois gametas haploides.

Na computação evolutiva, a combinação de dois "indivíduos" é geralmente chamada de *crossover*, que não é idêntico ao processo que ocorre nos organismos diploides, chamado de *crossing-over*. O *crossing-over* na verdade não ocorre na fertilização e sim durante a formação dos gametas, junto ao processo denomiado de *meiose*. A meiose é um processo de divisão celular que garante que os gametas terão apenas uma cópia de cada cromossomo, ou seja, serão haploides.

Como vimos anteriormente, Mendel foi o primeiro a entender o processo de hereditariedade nos organismos diploides. Passados vários anos, muito foi adicionado sobre a teoria inicial, mas ainda hoje não se entende por completo a teoria genética. O que sabemos é que toda a vida na terra está baseada no DNA (***d****eoxyribo****n****ucleic* ***a****cid*, ácido desoxirribonucleico em português) que é um composto orgânico, com a famosa estrutura em dupla hélice descoberta por Watson e Crick [**?**], cujas moléculas contêm as instruções genéticas que coordenam o desenvolvimento e funcionamento de todos os seres vivos. Os segmentos de DNA que contêm a informação genética são denominados *genes*. Para se obter as proteínas através do DNA passa-se por dois grandes processos. Primeiramente a informação do DNA é "escrita" no RNA, processo conhecido como *transcrição*. Depois temos a *tradução* que leva a informação do RNA até as proteínas, Fig. 3.4.

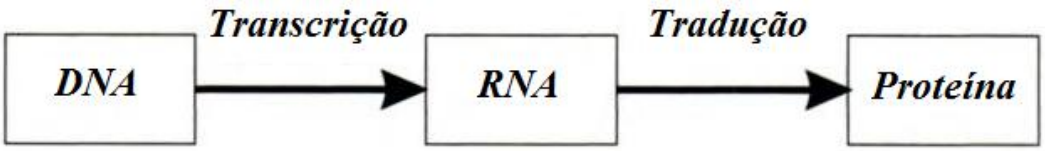



Figura 3.4: Processo que inicia-se no DNA até chegar as proteínas passando pela transcrição e tradução.

Hoje é amplamente aceito na genética molecular que a informação na Fig. 3.4 flui em apenas um sentido, ou seja, em termos de genótipos e fenótipos isso significa que novos fenótipos não podem alterar a informação genética. Esse fato refuta totalmente a teoria do uso e desuso proposta por Lamarck, que tinha como hipótese que características adquiridas durante a vida de um indivíduo poderiam ser transmitidas hereditariamente. Portanto, todo e qualquer tipo de variação e diversidade deve aparecer no nível microscópico através do mecanismo da mutação e recombinação, sendo depois testadas macroscopicamente pelo ambiente através da Seleção Natural.

## 3.3 Computação Evolutiva

O desenvolvimento de novas tecnologias é um dos objetivos da ciência atual. É neste contexto que a Natureza sempre serviu de fonte de inspiração para engenheiros e cientistas na busca da solução dos seus problemas. Duas grandes inspirações são o cérebro humano e o processo de evolução – que foi capaz de criar o cérebro humano. A primeira gerou o desenvolvimento do campo de *neurocomputação*, a segunda criou o objeto de estudo dessa dissertação, a computação evolutiva.

Um bom exemplo de aplicação da computação evolutiva na otimização de desenho industrial é o estudo feito do desenho de uma estrutura que liga uma antena de comunicação a um satélite [**?**]. Nesse problema queremos que esse suporte seja estável, principalmente com relação a vibrações, pois como não há ar no espaço para proporcionar uma resistência a essas vibrações, qualquer oscilação pode levar perigo à estrutura do satélite. Nesse trabalho Keane et al. otimizaram a estrutura do suporte empregando um Algoritmo Genético, e conseguiram obter uma estrutura que é 200 vezes melhor, com relação ao amortecimento das vibrações, que o desenho tradicional, Fig. 3.5.

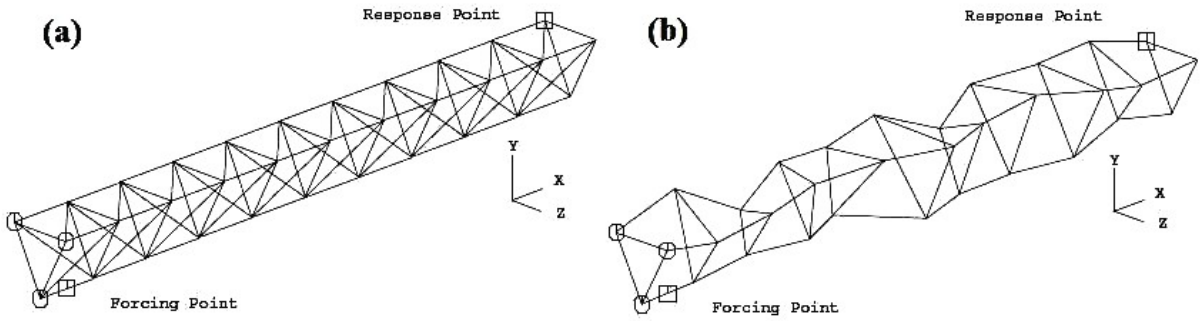



Figura 3.5: Processo de otimização de uma estrutura que liga a antena de comunicação de um satélite ao seu corpo. Um Algoritmo Genético foi empregado para modificar a geometria da estrutura alterando as 3 coordenadas das junções. O objetivo era minimizar a média de transmissão da vibração ao longo da estrutura. (a) Estrutura inicial, desenho regular da estrutura em 3D. (b) Desenho final da estrutura encontrado usando o Algoritmo Genético que chega a ser 200 vezes melhor que a estrutura regular [**?**].

A imagem da estrutura otimizada mostra uma característica que à primeira vista parece incomum, pois não apresenta nenhuma simetria e não há nenhum desenho lógico aparente. Isso ilustra o potencial de um algoritmo de busca como o evolutivo, pois ele é desprovido de preconceitos preexistentes. Dificilmente um engenheiro pensaria a princípio

que uma estrutura tão irregular quanto a vista na Fig. 3.5b poderia ser a estrutura ótima, ou pelo menos, a busca por exaustão seria um processo impossível, devido ao número enorme de soluções possíveis. É interessante notar que, nesse caso, o que torna a estrutura adequada é a sua própria assimetria. As ondas se propagam sobre a estrutura através das astes com diferentes tamanhos. Para uma dada combinação dessas diferenças, é possível que as ondas se anulem pelas diferenças de fases geradas. Esse representa um exemplo de um trabalho em que um algoritmo evolutivo foi utilizado como uma estratégia de otimização, mas de modo geral, podemos dividir os problemas em três classes: otimização, modelagem e simulação, como mostrada a Fig. 3.6. Para definir cada uma das classe, temos que levar em conta que um sistema físico pode ser teoricamente descrito em termos de três principais componentes: os *inputs*, o *modelo físico* e os *outputs*. Conhecer o modelo significa saber como o sistema funciona, ou melhor, saber quais são as leis que governam esse sistema. Os *inputs* seriam os parâmetros iniciais, que serão computados pelo modelo nos fornecendo como saída os resultados, ou os novos valores dos parâmetros.

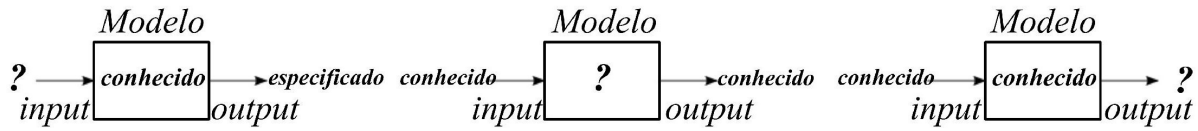



Figura 3.6: Três classes possíveis de problemas. *Esquerda.* Problema de otimização em que conhecemos o modelo e especificamos um *output*, mas não conhecemos o *input*. *Centro.* Problema de modelagem onde conhecemos os *inputs* e *outputs* mas não o modelo envolvido. *Direita.* Problema de simulação onde conhecemos os *inputs* e o modelo mas desejamos saber qual será o *output*.

O termo otimização se refere a situação na qual conhecemos o modelo junto com uma saída desejada, e queremos saber qual é a entrada, ou o conjunto de valores dos parâmetros que levam àquela saída desejada. O trabalho citado acima, sobre a estrutura nos satélites, trata de um problema de otimização. Conhecemos o modelo, na forma de equações que regem a propagação das ondas sobre o material, e procuramos uma solução que minimize a propagação das vibrações. A partir desse ponto, buscamos qual a estrutura inicial (*input*) que gera o mínimo de vibrações.

Já no caso da modelagem, os parâmetros de entrada são conhecidos assim como as suas saídas, o que não se conhece são as equações, ou o modelo que rege o sistema. A tarefa agora é encontrar qual o modelo que leva as entradas ao valores medidos das saídas.

Finalmente, na simulação conhecemos o modelo e os valores de entrada, mas buscamos saber os valores de saída correspondentes às entradas dadas. Hoje, com a existência de bons computadores, as simulações ganharam grande importância na ciência. As simulações nos permitem estudar sistemas cujos experimentos são raros e muito caros, ou até mesmo não existem, como é o caso da área de astrofísica ou a física de altas-energias.

## 3.4 Algoritmo Genético

Depois dessa introdução geral, podemos estudar de modo mais detalhado o Algoritmo Genético que será o algoritmo utilizado no nosso trabalho. Não há uma definição rigorosa, aceita por toda a comunidade, sobre o que ele seria e o que realmente o diferencia dos outros métodos de computação evolutiva [**?**]. O que podemos constatar é que a maioria dos Algoritmos Genéticos apresentam pelo menos quatro elementos em comum: população de cromossomos, seleção de acordo com *fitness*, crossover e mutações aleatórias.

A escolha da representação é um dos primeiros problemas que se enfrenta ao tentar resolver um problema utilizando Algoritmo Genético. A representação é como serão escritos os elementos da população no algoritmo. O modo mais comum é na forma de um vetor com elementos 0 e 1 (e.g. 000101100011), chamada de representação binária. Esse tipo de representação vetorial permite que se apliquem os operadores de crossover e mutação de modo mais natural, sendo por esse motivo o modo padrão de se representar soluções.

O operador de crossover é aplicado quando desejamos obter um novo indivíduo a partir de dois indivíduos que podemos chamar de "pais". Para a representação binária, podemos ter vários tipos de crossover. O *crossover de um ponto* foi o primeiro operador de recombinação proposto por Holland em 1975 [**?**]. Ele funciona quando sorteamos aleatoriamente um ponto dos vetores *pais*, dividindo-os em duas partes, uma à esquerda do ponto escolhido e outra região à direita. Dessa maneira, unindo a região esquerda de um vetor com a direita do outro, podemos construir um novo vetor. O mesmo pode ser feito com as regiões complementares, nos permitindo formar um segundo vetor "filho", como demonstra a Fig. 3.7a.

Um segundo tipo de crossover seria o *crossover de* **n**-*pontos*. Trata-se de uma extensão do crossover de um ponto, mas agora escolhemos $\boldsymbol{n}$ pontos nos vetores pais ao invés de apenas um. Escolhidos aleatoriamente esses $\boldsymbol{n}$ pontos, prosseguimos como no caso de um ponto, ou seja, alternamos entre as regiões entre os dois vetores construindo dois novos vetores filhos compostos por partes dos pais (ver Fig. 3.7b).

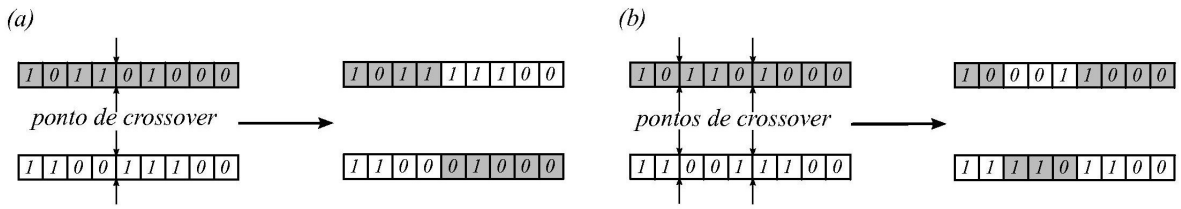



Figura 3.7: Dois possíveis tipos de crossover. Em *(a)* mostramos a criação de dois cromossomos "filhos" a partir do crossover de um ponto. Em *(b)* mostra a criação de dois cromossomos "filhos" a partir do crossover de $n$-pontos para $n$= 2.

Há muitas maneiras de aplicar o operador de mutação [**?**], porém, o modo mais simples e comum é varrer o cromossomo, quando representado por um vetor, modificando um certo "alelo" com uma pequena probabilidade $m$. No caso da representação binária, poderíamos ter inicialmente um vetor 100001001, e caso a terceira posição fosse sorteada, teríamos o novo vetor mutado como 10**1**001001.

O último operador sobre o qual vamos falar é o operador de seleção. O modo mais simples de aplicá-lo é selecionando uma porcentagem da população total para sobreviver até a proxima geração, e gerando todos os indivíduos que foram "mortos" nessa etapa. Ou seja, no caso mais simples, temos uma população de tamanho constante, digamos $P$ indivíduos. Depois de avaliados os *fitness* de todos eles, selecionamos apenas os $\lambda P$ ($0 < \lambda < 1$) indivíduos de maior *fitness*. Os selecionados devem gerar, através do crossover, $(1-\lambda)P$ indivíduos, que junto com os $\lambda P$ melhores formaram uma nova população de $P$ indivíduos, definindo uma geração.

A princípio, não levamos em consideração de que modo escolhemos a partir do grupo $\lambda P$ quais os pares que irão gerar a nova população. Isso pode ser feito de duas maneiras. Podemos sortear de modo uniforme, ou seja, todos os $\lambda P$ tem igual probabilidade de serem sorteados para gerar os $(1-\lambda)P$ pares que irão completar novamente a população. De outro modo, podemos introduzir uma distribuição de probabilidade de acordo com o *fitness* de cada indivíduo. O método chamado de *seleção proporcional a fitness*, introduzido em [**?**], estabelece que cada indivíduo $i$, com fitness $f_i$, tem uma probabilidade de ser selecionado igual a $f_i / \sum_j f_j$ ($j$ = 1 até $P$) . Podemos entender essa probabilidade como uma *fertilidade*, em que os indivíduos com maior *fitness* terão maior chance de reproduzir. Essa abordagem pretende fazer com que a seleção múltipla dos melhores para a reprodução nos leve a uma diminuição no número de gerações necessárias para encontrar regras eficientes. Isso pode introduzir alguns problemas; como a rápida diminuição da

diversidade, podendo o sistema ficar "preso" em algum mínimo local. Outro possível problema seria que, logo após poucos passos, a distribuição de *fitness* torna-se bastante regular entre os sobreviventes. Nessa situação a probabilidade de seleção teria pouca influência, tornando-se um gasto computacional desnecessário.

Tendo definido com precisão qual a representação para as possíveis soluções e uma função de *fitness* para a avaliá-las, podemos construir como seria o esquema de um possível Algoritmo Genético (ver Fig. 3.8):

**1.** Inicializa-se o programa gerando uma população inicial de $P$ (e.g., $P = 100$) indivíduos, cada um com *l-bits* escolhidos de modo aleatório.

**2.** Calcula-se o *fitness* de cada um dos indivíduos da população.

**3.** Constrói-se um ranque dos $\lambda P$ (e.g. $\lambda = 0.2$ então $\lambda P = 20$) melhores cromossomos, para gerarem a nova população.

**4.** Caso o maior *fitness* seja maior ou igual a meta a ser atingida, o programa é interrompido e imprime-se o melhor cromossomo. Caso não, passa-se para o passo **5**.

**5.** Escolhe-se $(1 - \lambda)P$ (e.g., $(1 - \lambda)P = 80$) pares retirados aleatoriamente do grupo selecionado em **3**, aplica-se o crossover sobre eles gerando $(1 - \lambda)P$ novos elementos e, logo após, aplica-se a mutação. Desse modo, gera-se novamente uma população completa de $P$ elementos.

**6.** Volta-se para o passo **2**.

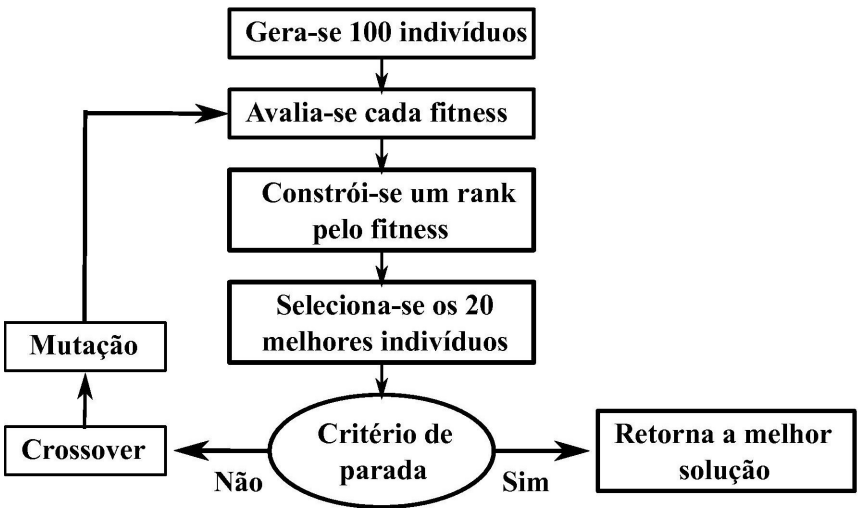



Figura 3.8: Fluxograma mostrando como é o esquema geral do funcionamento de um Algoritmo Genético, para o caso $P = 100$ e $\lambda = 0.2$.

## 3.5 Evolução de Autômatos Celulares

Um dos mais importantes exemplos teóricos de aplicação do Algoritmo Genético é a evolução de autômatos celulares com o propósito realizar computação. Essa aplicação está presente principalmente nos trabalhos de James Crutchfield [**?**], Rajarshi Das [**?**], Peter Hraber [**?**] e Melanie Mitchel [**?**]. Esse problema é amplo e pode ser visto tanto como um problema de otimização quanto de modelagem. Uma motivação para tal é entender como a evolução natural criou sistemas com *computação emergente*. Como a ação de componentes simples com acesso limitado a informação e comunicação pode produzir coordenação global e processamento de informação? Todos os exemplos citados no Capítulo 1 se enquadram nesse padrão: colônia de insetos, sistemas econômicos, sistemas imunes e mesmo o cérebro podem ser modelados através de autômatos celulares em que as regras específicas são obtidas através de algoritmos evolutivos. Por esse motivo, anteriormente mencionamos que os autômatos junto a algoritmos evolutivos podem ser utilizados para modelagem.

Nos anos 1940, Stanislaw Ulam estudou no Laboratório de Los Alamos o crescimento de cristais usando um modelo simples de uma rede regular bidimensional. Enquanto isso seu colega, John Von Neumann, estudava o problema de sistemas auto-replicantes, tentando primeiro utilizar um modelo em que um robô deveria construir outro, mas logo percebeu as dificuldades em projetar tal modelo. Ulam, então, sugeriu a Neumann que utilizasse de abstração matemática, e construisse um modelo parecido ao utilizado por ele no problema de crescimento de cristais. Desse modo, nasceu o primeiro autômato celular, uma rede bidimensional de "células" com acesso apenas a uma pequena vizinhança e tendo 29 possíveis estados [**?**, **?**, **?**].

Ainda nos anos 1940, Norbert Wiener e Artur Rosenblueth desenvolveram um modelo de autômato celular com o objetivo de fornecer uma descrição matemática para a condução de impulsos no sistema cardíaco [**?**].

Em 1969, o pioneiro da computação Konrad Zuse publicou seu livro *Rechnender Raum* ("Calcular o Espaço") [**?**], propondo que as leis da física eram discretas por natureza, e que o universo inteiro era o *output* de um computador determinístico em um grande autômato celular. Seu livro é considerado como o primeiro livro sobre o que chamamos hoje de *física digital*.

Em 1970, um autômato celular de dois estados chamado "Jogo da Vida" ficou bastante famoso. Foi inventado por John Conway e popularizado por Martin Gardner na sua coluna da *Scientific American* sobre jogos matemáticos [**?**, **?**, **?**]. As regras são bem simples,

podemos ter dois estados possíveis (e.g. *preto* ou *branco*, que poderiam simbolizar vivo ou morto), e cada estado é atualizado no tempo de acordo com os estados do seus vizinhos. Caso uma célula tenha 2 vizinhos pretos ela se mantém como está, se ela tem 3 vizinhos pretos ela se torna preta. Para todas as outras possibilidades, o próximo estado deve ser branco. Apesar dessa simplicidade nas regras, o sistema apresenta uma grande diversidade de comportamentos gerando um grande interesse no seu estudo.

Stephen Wolfram em 1982 publicou o primeiro de uma longa série de artigos, analisando sistematicamente propriedades básicas, mas essenciais, de uma classe de autômatos [**?**, **?**, **?**, **?**]. O comportamento complexo inesperado dessas regras simples levaram Wolfram a suspeitar que a complexidade na natureza pode aparecer por mecanismos similares. Em 2002, Wolfram publicou o livro *A new Kind of Science*, que se tornou uma das referências mais citadas na área [**?**].

Com os trabalhos do Wolfram e com o sucesso do Jogo da Vida, a área de complexidade e autômatos celulares virou uma grande febre nos anos 80, mas o que poucos sabem é que até físicos teóricos famosos como Richard Feynman tentaram realizar pesquisas nessa área. Feynman foi convencido por um engenheiro e matemático chamado W. Daniel Hillis, cofundador de uma empresa chamada *Thinking Machines Corporation*, cujo objetivo era construir um supercomputador para processamento paralelo. Hillis conta a história de como teve a colaboração de Feynman no projeto, num artigo publicado em homenagem a Feymann logo após sua morte [**?**]. Neste artigo, Hillis conta que uma das primeiras rotinas utilizadas para testar o funcionamento do computador foi um algoritmo para o Jogo da Vida, isso porque um autômato celular pode ser visualizado como uma máquina de processamento paralelo. Hillis junto a Feynman também estudaram estratégias evolutivas, chegando até a produzir um resultado que corroborava a *teoria do equilíbrio pontuado*, mas que depois Hillis verificou já ter sido publicado. A *teoria do equilíbrio pontuado* diz que mudanças evolutivas ocorrem de forma rara e localizada em eventos rápidos de especiação e não de modo gradual como defende a teoria do *gradualismo*. Hillis conta que ao telefonar para Feynman e lhe contar que tudo que eles haviam feito já havia sido publicado, Feynman ficou muito entusiasmado: “Acertamos no alvo! ”, exclamou ele, “Nada mau para amadores.” [**?**].

O autômato celular mais simples que podemos estudar é unidimensional com dois estados possíveis. O conjunto de cada estado $s_i$ de todas as células $i$ é chamado de configuração do autômato. Ou seja, temos $N$ células ligadas com seus $k$ vizinhos mais próximos (incluindo ele mesmo). Cada célula pode estar em um de dois estados, que é

atualizado em passos discretos de tempo segundo uma regra, ou função $\phi$. Essa regra pode ser visualizada na forma de uma tabela, como mostrado na Fig. 3.9. A cada passo de tempo, percorremos o autômato varrendo todas as células, e para cada célula é construído um padrão de *input*, com os valores dos estados dos seus vizinhos. De posse do *input*, procura-se na tabela da regra $\phi$ qual é o *output* respectivo. Repete-se esse processo, de modo sincronizado, para todos os estados do autômato com o objetivo de se obter uma nova configuração.

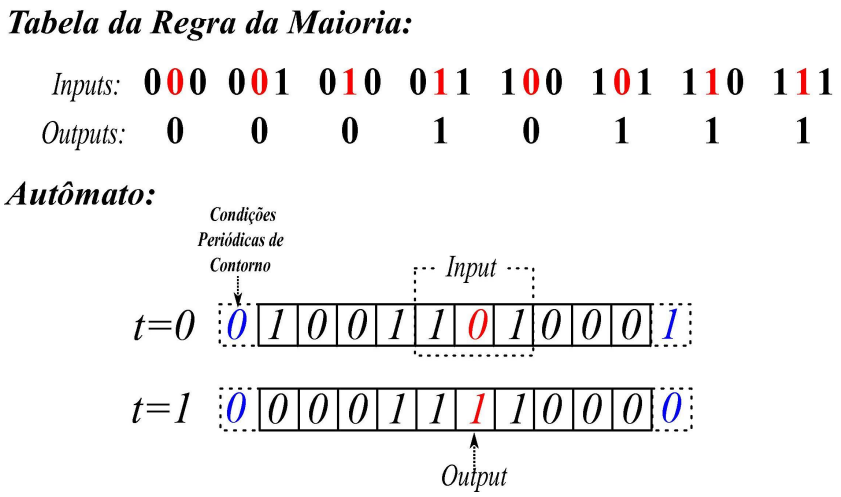



Figura 3.9: Tabela para a regra da maioria e atualização dos estados de um autômato. Acima, vemos exemplificado uma das possíveis regras, a chamada regra da maioria. Nela o *output* é dado pelo valor que mais se repetir naquele padrão de *input*. Logo abaixo, é mostrado um conjunto de células com seus estados formando a configuração inicial do autômato ($t = 0$). Buscamos ressaltar como se determina a vizinhança, e consequentemente o padrão de *input* para aquela célula, assim como o fato do autômato obedecer condições periódicas de contorno. Cada padrão de *input* é medido e comparado com a tabela para se obter o respectivo *output*, e assim, atualizar o autômato para a configuração de $t = 1$.

No caso de um autômato de dois estados, podemos nos utilizar da representação binária para construir a tabela na forma de um vetor. Cada índice do vetor representará um *input* diferente. O número de *inputs* possíveis é dado por $2^k$, pois cada vizinho pode ter dois estados e temos $k$ vizinhos. No caso do exemplo da Fig. 3.9, temos a representação da *regra da maioria* para $k = 3$, ou seja, cada célula tem acesso ao seu estado e ao estado dos seus dois vizinhos mais próximos, um a direita e outro a esquerda. Portanto, devemos ter um vetor de tamanho $2^3 = 8$ para guardar a nossa regra. A regra da maioria é somente uma das $2^{2^k}$ (e.g. para $k = 3$ temos $2^{2^3} = 256$) possíveis regras. Nela cada estado é

atualizado de acordo com o total de estados 1 ou 0 dos seus vizinhos, mais precisamente, o *output* é 0 caso a maioria de vizinhos seja 0, e 1 caso a maioria seja 1.

Do mesmo modo que os *inputs* podem ser entendidos como representações binárias de números inteiros, as regras também podem ser vistas como números inteiros. Ao invés de termos que citar um vetor ou tabela a todo momento que nos referimos a uma regra específica, podemos usar o *código de Wolfram*, que nada mais é que o inteiro representado pelo número binário do vetor da regra. Ou seja, a regra da maioria para $k = 3$ é **00010111**, que é um número binário que transformado para representação decimal fica $0 * 2^7 + 0 * 2^6 + 0 * 2^5 + 1 * 2^4 + 0 * 2^3 + 1 * 2^2 + 1 * 2^1 + 1 * 2^0 = 23$. Então, a regra da maioria é a **regra 23**.

A dinâmica de um autômato celular – quando falamos em autômato não nos referimos apenas à configuração do sistema de células, mas também à regra que gera aquela configuração – é normalmente representada por um diagrama, em que se ordena as diferentes configurações obtidas para cada passo de tempo, e os estados são, ao invés de 0 e 1, cores como preto e branco. Na Fig. 3.10 vemos o comportamento de 3 regras diferentes para um autômato unidimensional com $k = 3$. Na Fig. 3.10a, temos a regra **184** que é utilizada para modelar tráfico de veículos [**?**], e para problemas de aniquilação balística onde as interfaces de 0 e 1 podem ser vistas como partículas se movendo e colidindo [**?**]. Na Fig. 3.10b temos o comportamento da regra **110**, famosa pelos trabalhos de Wolfram e Matthew Cook [**?**, **?**].

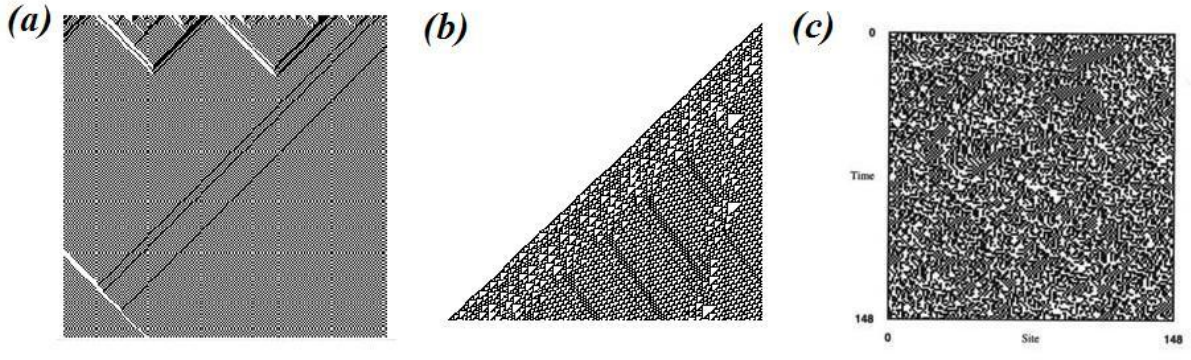



Figura 3.10: Diagrama de configurações para três diferentes regras, onde o tempo é representado pelo eixo vertical. *(a)* Diagrama para a evolução da regra **184** para uma condição inicial aleatória. Essa regra é utilizada para modelar tráfico de veículos. Em *(b)* temos a regra **110**, famosa por ser uma máquina de Turing universal, aplicada a uma condição inicial de apenas uma célula diferente de 0. Em *(c)* temos uma regra produzida aleatoriamente aplicada em uma condição inicial também aleatória.

Na Fig. 3.10c, temos o desenvolvimento de uma regra gerada aletoriamente. Podemos ver que não há um padrão bem formado no desenvolvimento temporal do autômato. Esse padrão aleatório está presente na maioria das regras. Poucas regras são capazes de realizar algum tipo de computação paralela mais complexa. Por computação, entendemos a capacidade de uma regra de extrair informações sobre qualquer configuração inicial através da formação de padrões durante a dinâmica do autômato. Sendo assim, o Algoritmo Genético torna-se uma ferramenta útil na busca desse tipo de regra que realiza alguma computação não trivial.

## 3.6 Problema da Maioria

Uma possível tarefa computacional que podemos impor ao nosso autômato é o chamado *problema da maioria*, ou *classificação por densidade*. Essa tarefa computacional modela o problema da coordenação global presente em sistemas complexos, especificamente serve como uma medida de coordenação e processamento global de informação [**?**]. De tal maneira que, dado um sistema composto de muitas células interligadas localmente, cada uma com uma variável de estado binária, pode-se dizer que o sistema resolve o problema da maioria, caso todas as células convirjam para o mesmo estado, e o sistema atinja uma configuração idêntica ao estado que era maioria na configuração inicial. Se utilizarmos um autômato como modelo, podemos dizer que, caso tenhamos uma configuração inicial com maioria 1, então uma regra que realize a classificação por densidade deverá convergir todos os estados para 1. O problema da maioria é trivial se pensarmos em um sistema com controle central, bastaria apenas realizar uma contagem dos estados e enviar a informação a todas as células que mudem seu estado para o estado correto. Por outro lado, no caso de um sistema descentralizado, a tarefa é não-trivial, pois o sistema necessita transpor dois desafios: (i) extrair informação global de um padrão apenas local e (ii) coordenação global, convergindo todos os estados para o mesmo estado [**?**].

Mais formalmente, podemos ter a densidade $\rho$ de 1's na configuração do nosso autômato em um dado passo de tempo. Dessa maneira, temos que $\rho_c = 1/2$ seria uma densidade crítica para a classificação. Sendo $\rho_0$ a densidade na configuração inicial, devemos ter que, antes de um tempo limite $T$, caso $\rho_0 > \rho_c$, o autômato deve atingir uma configuração uniforme de estados iguais a 1. O contrário deve ocorrer para $\rho_0 < \rho_c$. Para um tempo limite $T$, devemos ter todos os estados iguais a 0. O tempo limite é de certo modo um parâmetro do problema, normalmente exigido como uma função linear do tamanho do sistema $N$, empregando-se usualmente $T = 2N$ [**?**, **?**].

O primeiro candidato a se pensar para solucionar esse problema seria a regra da maioria. Para testar essa hipótese, Mitchell construiu o diagrama das configurações para a regra da maioria com $k = 7$ para os casos $\rho_0 < 1/2$ e $\rho_0 > 1/2$, como mostrado na Fig. 3.11. Como podemos ver, existe a formação de domínios estáveis, pois as células das interfaces entre as regiões com 0's e as regiões com 1's são incapazes de decidir entre os dois estados. Consequentemente, a regra da maioria não é capaz de resolver o problema da classificação de densidade.

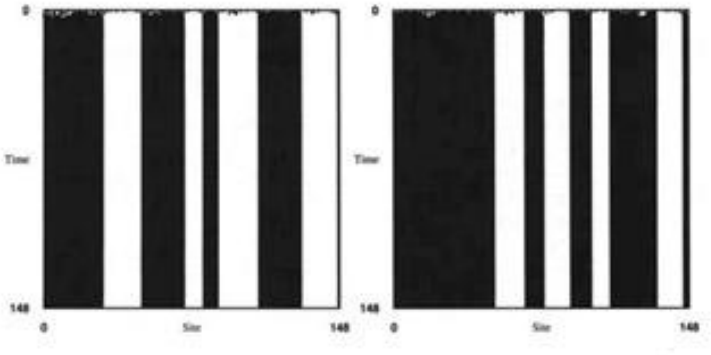


Figura 3.11: Diagrama de configurações para a regra da maioria com $k = 7$. O eixo vertical representa o tempo e o horizontal os estados de cada célula. A *esquerda* temos o caso em que $\rho_0 < 1/2$. A *direita* temos o caso em que $\rho_0 > 1/2$. Vemos que em ambos os casos a regra da maioria não leva o sistema a convergir para uma configuração de consenso (retirado de [**?**]).

Devemos, então, buscar um modo de procurar regras capazes de realizar essa tarefa. Nesse ponto, Crutchfield e Mitchell aplicaram o Algoritmo Genético como método de busca [**?**]. O primeiro passo foi estabelecer a representação da solução. Os autores construíram o vetor solução com os *outputs* da tabela de cada regra, listados em uma ordem *lexicográfica* dos vizinhos, assim como na Fig. 3.9. Para o caso $k = 7$, temos um vetor regra de tamanho $2^k = 128$. O espaço das possíveis soluções tem $2^{128}$ regras, número da ordem de $10^{38}$. Este valor é tão grande que seria impossível realizar uma busca por exaustão. Se aumentarmos mais dois vizinhos, teremos $k = 9$, cujas regras tem tamanho $2^9 = 512$. Para $k = 9$ teríamos um número de $2^{512}$ regras diferentes, da ordem de $10^{154}$, bem maior que o número estimado de átomos no universo ($10^{80}$ átomos [**?**]). Isso mostra a dificuldade de encontrar regras que realizem a classificação por densidade, principalmente quando levamos em conta que apenas uma pequena parte desse conjunto enorme é de regras eficientes.

Nos trabalhos [**?**, **?**] foram usados sistemas com tamanho $N = 149$ e o Algoritmo

Genético começava com uma população gerada aleatoriamente. Na verdade, a evolução iniciada a partir de regras completamente aleatórias para $k = 7$ é um processo lento e exige muitas gerações. A justificativa para isso já foi mencionada no parágrafo anterior, pois o número de regras possíveis com $k = 7$ é muito grande e investigar esse espaço sem nenhuma informação *a priori* é uma tarefa exaustiva. O que pode ser feito é introduzir uma distribuição não uniforme quando o algoritmo for gerar as primeiras regras. Isso faz com que nem todas as regras apareçam inicialmente com a mesma frequência, acelerando a convergência [**?**]. Considerando esta abordagem, Crutchfield e Mitchell no trabalho [**?**] propõem iniciar o algoritmo com 100 regras. O próximo passo é avaliar o *fitness* de toda essas 100 regras. Para essa tarefa, os autores escolheram aleatoriamente 100 condições iniciais (CI) para os autômatos, ou seja, 100 distribuições de 0's e 1's. Essa escolha também pode ser feita de dois modos. Podemos fixar uma densidade $\rho_0$, como por exemplo $\rho_0 = 0.6N$ e partindo disso cria-se aleatoriamente 50 autômatos com $\rho_0$ e 50 com $1 - \rho_0$, ou seja, devemos ter metades das CI com maioria 1 e a outra metade com maioria 0. Isso deve ocorrer, pois é muito fácil achar uma regra que tenha *fitness* máximo, caso somente haja maioria 1. Basta imaginar a regra **1111111**, ela automaticamente altera todos os estados para 1, de fato não realizando nenhum tipo de computação. O mesmo ocorreria, caso a maioria sempre fosse 0. Quando temos apenas a metade das CI's para cada um dos possíveis *bits*, impomos que essas regras tendenciosas tenham um *fitness* máximo de 50%. Outro modo de se gerar CI's é através de uma distribuição uniforme para $\rho_0 \in [0.0, 1.0]$, mas, também, tomando cuidado para que metade seja com maioria 1 e a outra metade 0. Levando esse fato em conta, os autores aplicaram a cada regra às 100 diferentes CI's aleatoriamente geradas, e observaram em quantas delas, em um tempo menor que $2N$, o consenso foi perfeitamente atingido. A fração dessas condições em que houve a classificação perfeitamente correta fica gravada como o *fitness* $f_{100}$ dessa regra. Vale notar que nenhum crédito parcial é dado, deveriam ter exatamente $N$ estados 1 ou 0 dependendo apenas de $\rho_0$.

Portanto, a cada geração o algoritmo deve executar os seguintes passos [**?**, **?**]:

(i) Gera-se 100 CI's para cada regra;

(ii) $f_{100}$ é calculado para cada regra na população;

(iii) Constrói-se um rank em ordem decrescente do *fitness*, e as 20 melhores são copiadas para a próxima geração;

(iv) As outras 80 regras são geradas através de crossover de um ponto entre as 20 melhores, e é aplicado mutação nessas regras.

Observou-se que a evolução através do AG ocorre por sucessivas épocas de inovação. Cada uma dessas épocas é marcada pela descoberta de uma nova estratégia que melhora a performance, aumentando o *fitness* da melhor regra, como mostrado na Fig. 3.12. Depois de atingido um patamar desejado no *fitness*, o desafio é interpretar como aquela regra realiza a tarefa com eficiência, pois evolutivamente não é sugerido nenhum tipo de estratégia, apenas é imposto, através da "pressão" de seleção, o aumento do *fitness* relativo. O mesmo ocorre com a evolução natural, ou seja, na dificuldade que os biólogos têm para compreender todas as características de animais e o porquê do funcionamento delas. No caso das regras dos autômatos, a investigação ocorre pela observação do diagrama de configurações para diferentes CI's.

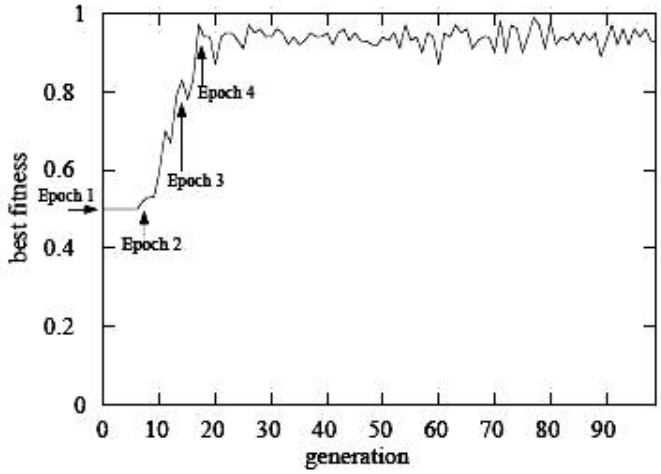



Figura 3.12: Maior *fitness* (*best fitness*) versus número de gerações (*generations*) em um processo típico de evolução para um autômato celular. As quatro épocas (*epochs*) de inovação estão indicadas, cada uma correspondendo à geração em que há a descoberta de uma nova estratégia mais eficiente (retirado de [**?**]).

O diagrama de configurações nos permite visualizar como uma regra específica realiza a computação da informação inicial de uma dada maioria de estados. São realizadas várias execuções do algoritmo genético, pois, como se trata de um algoritmo estocástico e como o *landscape* das soluções possui vários máximos locais, é de se esperar o aparecimento de diferentes estratégias em diferentes execuções. Nos trabalhos [**?**, **?**] foram encontradas duas regras capazes de realizar a classificação por densidade, com *fitness* variando entre $0.60$ e $0,80$. As regras de mais baixo *fitness* encontradas foram as chamadas *expand-blocks*, uma classe de regras que segue uma estratégia não muito sofisticada e seu *fitness* diminui à medida que aumentamos o sistema [**?**, **?**].

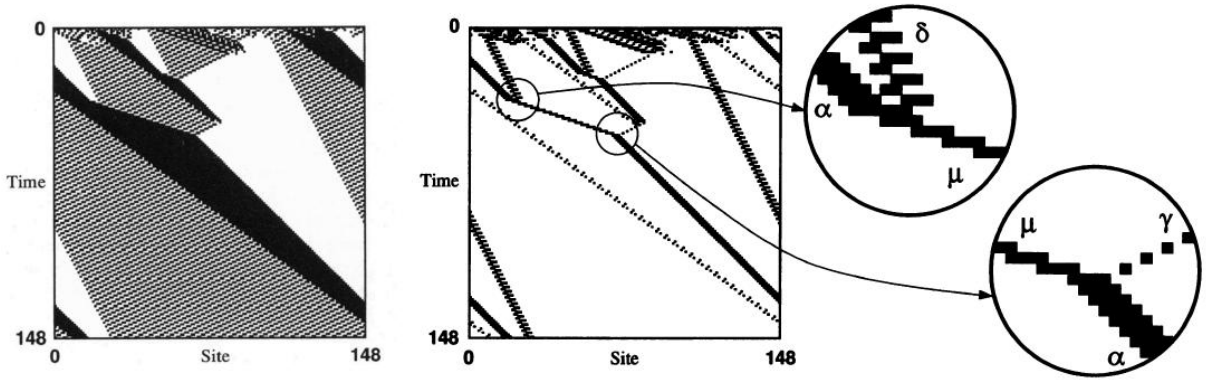



Figura 3.13: *Esquerda.* Diagrama de configurações para uma regra tipo-partícula apresentando domínios e interfaces. *Direita.* Mesmo diagrama, mas retirando todos os domínios para tornar mais clara a visualização das "partículas" e suas interações (retirado de [**?**]).

Uma segunda classe de regras encontradas, as denominadas regras *tipo-partícula*, realizam uma computação mais sofisticada que as regras *expand-blocks*. Elas têm *fitness* relativamente alto, variando entre 0.69 e 0.77 para sistemas com $N = 149$, e não variando muito com esse tamanho. Crutchfield e Hanson desenvolveram um método geral para entender a computação presente nos padrões dos diagramas de configurações em termos da formação e interação de domínios e "partículas" [**?**, **?**]. Observando os diagramas de configuração dessas regras, eles observaram que entre os domínios, regiões com um padrão regular e repetido de estados, há vários padrões de transições diferentes e que interagem entre si, dando a idéia de interações entre partículas, como ilustrado na Fig. 3.13. Desse modo, cada interface diferente entre domínios é identificada como uma partícula com propriedades específicas, e que podem interagir através de três processos: aniquilação, decaimento e reação, como mostrado na Fig. 3.14. Essas "partículas" carregam as informações locais que são processadas através das interações entre elas, permitindo que o sistema atinja uma coordenação global.

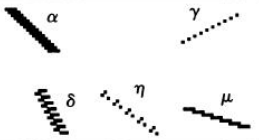
| | Particle | | | Particle interaction (by type) | | |
|---|---|---|---|---|---|---|
| Domain | Symbol | Velocity | Graphics | Annihilation | Decay | Reaction |
| $\Lambda^0 = 0^*$ | $\alpha \approx \Lambda^1\Lambda^0$ | 1 |  | $\delta + \gamma \to \emptyset$ | $\beta \to \delta + \eta$ | $\alpha + \delta \to \mu$ |
| $\Lambda^1 = 1^*$ | $\beta \approx \Lambda^0\Lambda^1$ | 0 | | $\mu + \eta \to \emptyset$ | | $\mu + \gamma \to \alpha$ |
| $\Lambda^2 = (10001)^*$ | $\gamma \approx \Lambda^2\Lambda^0$ | $-2$ | | | | $\eta + \alpha \to \gamma$ |
| | $\delta \approx \Lambda^0\Lambda^2$ | ½ | | | | |
| | $\eta \approx \Lambda^2\Lambda^1$ | ⅔ | | | | |
| | $\mu \approx \Lambda^1\Lambda^2$ | 3 | | | | |

Figura 3.14: Figura mostrando uma tabela retirada do artigo [**?**], que apresenta as configurações que representam cada tipo de "partícula" em termos das diferentes interfaces formadas entre os domínios. Na tabela, também podemos ver como essas "partículas" interagem entre si.

Durante esse capítulo buscamos explicar os detalhes da construção de um Algoritmo Genético, e por se tratar de um algoritmo simples preferimos focar a maior parte do texto em exemplos e aplicações. Buscamos mostrar que apesar da sua simplicidade o Algoritmos Genético é capaz de achar soluções sofisticadas e de comportamento complexo, como é o caso das regras tipo-partículas. Nosso trabalho será explicado com detalhes no próximo capítulo e veremos que ele tem como base os trabalhos de Crutchfield e Mitchell sobre a Evolução de Autômatos Celulares. Mostraremos que o Algoritmo Genético pode ir além do encontrado por Crutchfield e Mitchell, sendo capaz de encontrar regras muito maiores, com relação ao número de vizinhos, sendo necessário apenas o acréscimo de um algoritmo de promoção para o tamanho das regras.

# 4 Algoritmo genético na solução do problema da maioria

## 4.1 Externalidade na tomada de decisão

Nos anos 50 o psicólogo social Solomon Asch realizou uma série de famosos experimentos. Em um deles Asch dispôs grupos de 8 pessoas em uma sala com o propósito de observar um conjunto de 20 *slides* projetados sobre a parede. Nesses *slides* haviam 4 linhas verticais de diferentes tamanhos. Em casa *slide* havia um grupo distinto de 3 linhas com diferentes tamanhos, cada uma identificada por uma letra e ao lado das três linhas havia outra linha de tamanho idêntico ao de uma delas, como ilustra a Fig. 4.1. A medida que cada *slide* era projetado o pesquisador, que acompanhava o grupo de 8 pessoas testadas, perguntava para cada uma delas qual das 3 linhas indicadas tinha o mesmo comprimento da última. Devemos notar que os desenhos eram sempre feitos de modo que a solução era óbvia. O ponto chave do experimento estava no fato de que somente uma das 8 pessoas estava realmente sendo testada, enquanto que todas as outras eram atores que seguiam um protocolo de respostas previamente combinado.

Com esse experimento Asch desejava testar qual é a influência da opinião dos outros sobre a nossa tomada de decisão, ou seja, testar a idéia da economia clássica fundada por Adam Smith do homem racional que toma suas decisões avaliando o custo de modo egoísta e individual. Para testar essa hipótese primeiramente os 7 atores eram indicados a acertar os primeiros *slides*, dando assim confiança ao indivíduo testado. Logo após, quando o pesquisador questionava os atores em sequência, todos escolhiam uma mesma resposta, mas agora escolhiam a errada. A pessoa a ser testada era sempre a última a ser perguntada para que pudesse escutar a resposta de todos os atores e ser influenciada. O que Asch reportou em seu artigo foi que cerca de 1/3 das pessoas mudaram de opinião, preferindo seguir a resposta unânime dos atores, mesmo esta sendo claramente errada. Baseado nisso, o que Asch chamou de *experimento de conformidade*, trata-se do fato de

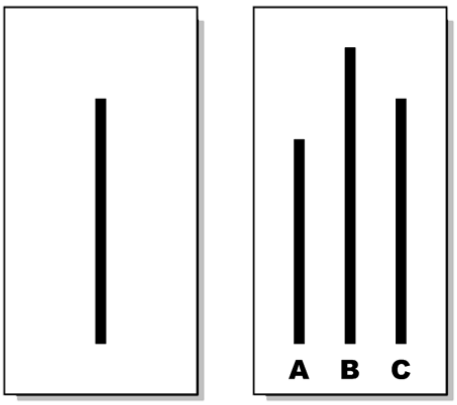



Figura 4.1: Experimento de conformidade de Asch. Na figura vemos um dos *slides* usados por Asch no seu famoso experimento de conformidade. Cada pessoa devia responder quais das retas a direita tem o mesmo tamanho que a reta da esquerda. As retas eram desenhadas de tal modo que a resposta era óbvia (nesse caso a resposta é a reta C). Asch tinha como objetivo testar se poderíamos mudar as nossas opiniões, ou pelo menos expressá-las de modo diferente, de acordo com a opinião de vizinhos.

uma pessoa ou um grupo estar em *conformidade* a algo estabelecido por outro grupo, comumente uma maioria [**?**].

De fato, a espécie humana é extremamente adaptada a vida em sociedade e para isso aprendeu a retirar o máximo de vantagem dessa condição. Muitas vezes nos deparamos com problemas muito complexos ou incertos para a nossa avaliação, e aprendemos que ao levar em consideração a opinião global temos mais chance de acertar ou mesmo de não nos colocar em uma situação de risco. Estamos sempre avaliando em que momento devemos aderir a uma nova tecnologia ou qual a hora menos "arriscada" de entrar em uma manifestação política. Do mesmo modo, podemos imaginar uma situação em que queremos almoçar e estamos em dúvida entre dois restaurantes, os dois sendo similares no que se refere a mesma decoração e preços, mas um deles tendo muito mais clientes que o outro. Qual dos dois devemos escolher? É fácil imaginar que apenas pessoas com problemas com multidões escolheriam o restaurante vazio. Esse é um exemplo simples em que seguimos a *regra da maioria*, ou seja, tomamos a decisão idêntica a tomada pela maioria. O mesmo comportamento pode ser encontrado em animais. Em experimentos de escolha de alimento, um rato irá continuar comendo uma dada comida se a maioria dos outros ratos também estiverem consumindo, mesmo se a comida estiver induzindo náuseas enquanto ele a come [**?**].

O *fenômeno da conformidade* não apenas ajuda na busca da solução de problemas

complexos, mas também diminui o número de conflitos em um grupo ou sociedade, pois os leva a um estado de coerência. A conformidade traz como consequência um estado de consenso e coordenação global ao grupo. Sabemos que essas propriedades são de extrema importância na sobrevivência de animais, como algumas espécies de pássaros e peixes que vivem em comunidades ou que adotam estratégias de grupo na fuga de predadores. Muito provavelmente o aparecimento e manutenção da conformidade foi favorecido evolutivamente devido à sua utilidade na sobrevivência desses animais.

Para entender precisamente como ocorre a coordenação global, criaram-se modelos de como as nossas decisões são afetadas pelas decisões dos nossos vizinhos. Imagine que estamos participando do experimento de Ash e seis pessoas escolheram como resposta a linha **C**, enquanto apenas uma delas respondeu que a linha de mesmo tamanho é a linha **B** (ver Fig. 4.1). O indivíduo que já tinha a sua opinião previamente formada de que a resposta certa é **C**, nessa situação irá ignorar a opinião do único que escolheu **B** e irá se manter junto à maioria e à sua convicção. Suponhamos agora que dois atores responderam **B**, muito provavelmente o indivíduo ainda seguirá a sua convicção respaldada pela opinião da maioria. Seguindo esse mesmo raciocínio, podemos ir aumentando o número de pessoas com opinião contrária à do indivíduo. O que se observa é que existirá um *limiar* onde este irá preferir expressar uma opinião diferente da sua convicção. No seu experimento, Ash observou que a maioria das pessoas necessitava que praticamente todos os atores respondessem contra elas para que mudassem de opinião, ou seja, bastava que houvesse um ator com opinião a favor para que os indivíduos testados se sentissem confiantes para expressar a sua opinião, mesmo contrária à maioria. Portanto, o limiar de mudança de opinião é próximo de 100% do grupo de vizinhos, mas esse limiar deve variar de situação para situação, assim como de pessoa a pessoa. Nesse caso temos um limiar alto, pois a resposta é bastante óbvia, sendo necessária uma "pressão" social muito grande para haver uma mudança de opinião. Já em situações de decisões mais delicadas teríamos um limiar mais baixo. Um gráfico da probabilidade de escolha de uma opção pode ser visto na Fig. 4.2, quando comparado com a probabilidade utilizada em modelos de infecção.

Em uma versão mais simples de um *modelo de limiar* teríamos uma população em que cada indivíduo tem um valor de limiar característico, gerando uma distribuição de frequência desses limiares dentro dessa população. Suponhamos como exemplo a situação de uma manifestação, logo cada indivíduo é confrontado com a decisão de entrar ou não nessa manifestação. Suponhamos, também, uma distribuição simples, na qual em uma população de 100 indivíduos temos um indivíduo com limiar 0, ou seja, ele participa da manifestação mesmo não tendo ninguém aderido ainda, temos outro com limiar 1 que

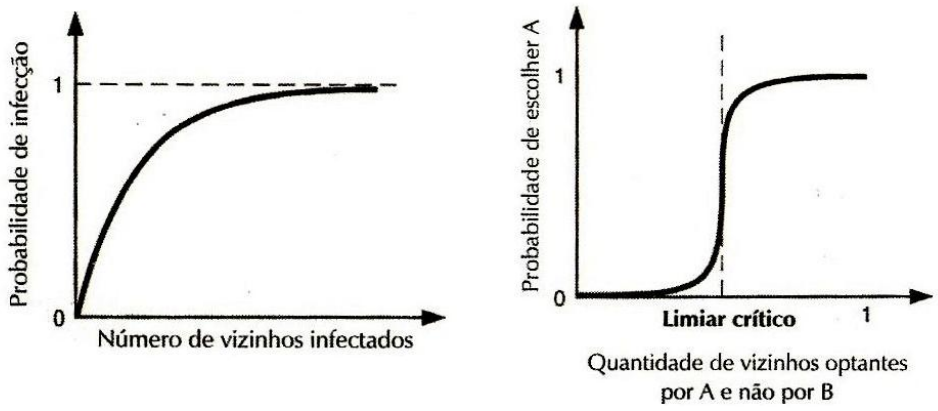



Figura 4.2: Comparação da probabilidade da mudança de estado para dois modelos de agentes diferentes. *Esquerda.* Temos a probabilidade de infecção em um modelo típico de propagação de doenças. Nele vemos que a a probabilidade de infecção sempre aumenta com o aumento do número de contatos com pessoas infectadas. *Direita.* Probabilidade de se escolher uma dada opção diante de um número de vizinhos optando por ela. À medida que aumentamos o número de vizinhos escolhendo a opção **A**, vemos que há um limiar crítico em que se muda da opção **B** para **A**. Retirado de [**?**].

aderirá logo após o indivíduo com 0 aderir, temos outro com limiar 2, e assim sucessivamente até o indivíduo com limiar 100. É fácil observar que essa distribuição irá levar à um efeito em cascata, onde todos irão aderir a manifestação, mas bastaria mudar o limiar de um dos indivíduos para um valor acima para estagnarmos o aumento do movimento de manifestação. Esse modelo pode ainda ser melhorado ao levarmos em conta outros fatores como a limitação espacial, pois cada indivíduo tem informação limitada, tendo acesso apenas à opinião de poucos vizinhos. O modelo de limiar, apesar da sua simplicidade, pode ser aplicado em diversas situações práticas [**?**].

1. Experimentos de psicologia social. Experimentos de conformidade, como o de Asch, dependendo do número de atores introduzidos [**?**, **?**]. Experimentos em que o espectador pode ser descrito por meio de um "limiar de ajuda".

2. Migração. Sabemos que a decisão de migrar depende fortemente de outros tomarem a mesma decisão, formando uma "cadeia de migração" [**?**].

3. Difusão de inovações. Mulheres em uma vila podem ser cautelosas ao adotarem meios contraceptivos e esperarem que uma dada proporção de vizinhas façam o mesmo [**?**].

4. Estudo de greves. Cada trabalhador para entrar em greve deve observar com cautela o número de trabalhadores que já aderiram, pois o custo de ser uma minoria nessa situação pode ser alto.

## 4.2 Modelo e Resultados

Modelamos o problema da coordenação global como uma tarefa computacional. Utilizamos o *problema da maioria*, também chamado de problema da classificação por densidade, como uma medida de coordenação global e processamento de informação [**?**, **?**]. O modelo adotado é baseado em um autômato celular (explicado com detalhes no capítulo anterior), no qual cada unidade pode ter apenas dois estados distintos, por exemplo 1 ou 0. Em cada passo de tempo, o estado de cada célula é atualizado segundo uma regra. Essa regra leva em consideração o fato de cada célula ter acesso à informação limitada, ou seja, cada célula conhece o estado de $k$ "vizinhos" (contando com o seu estado). Procuramos qual regra é capaz de resolver com eficiência o problema da maioria, cujo objetivo é evoluir o autômato a uma configuração em que todos os estados são iguais ao estado de maioria inicial. Adicionamos também uma limitação temporal à computação, pois esperamos que o número de passos para se atingir o estado de consenso seja menor que duas vezes o tamanho do sistema ($2N$, com $N$ igual ao número de células).

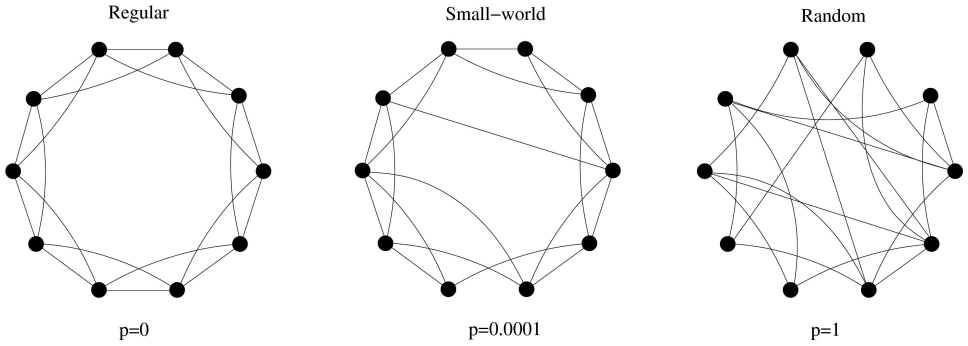



Figura 4.3: Modelo de rede de "mundo-pequeno" tal como introduzido por Watts e Strogatz [**?**, **?**]. Inicia-se com um rede regular com condições periódicas de contorno, como mostrado mais a esquerda. A partir desse ponto, redireciona-se cada ligação de acordo com uma pequena probabilidade $\rho$. Para $\rho = 0$ temos uma rede regular, ao aumentarmos esse valor, aumentamos a aleatoriedade das ligações chegando ao extremo $\rho = 1$ em que o sistema torna-se uma rede aleatória. Entre esses dois extremos, para uma pequena probabilidade, obtemos o fenômeno de mundo-pequeno.

Sabemos que as relações sociais não formam uma rede regular e sim uma rede complexa, logo para que o nosso modelo apresente esse importante ingrediente, utilizamos o modelo de "mundo-pequeno" introduzido por Watts e Strogatz [**?**, **?**]. Nesse modelo constrói-se primeiramente uma rede regular com condições periódicas de contorno, em que cada nó (célula no autômato) está ligado aos seus $k - 1$ vizinhos mais próximos e também ligado a si mesmo – completando $k$ ligações. Logo após isso, varremos todas as ligações da rede redirecionando essa ligação para um novo nó com uma pequena probabi-

lidade $\rho$, como mostrado na Fig. 4.3.

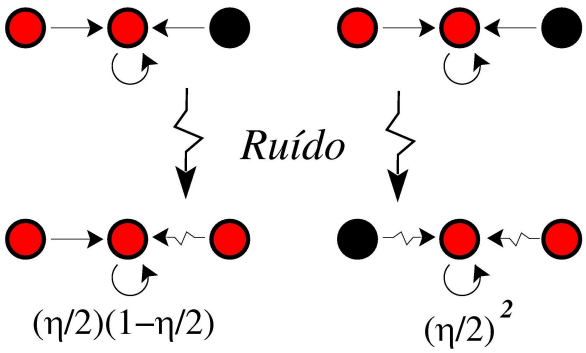



Figura 4.4: Descrição do efeito do ruído na informação recebida por uma célula e probabilidade de cada configuração. Para atualizar seu estado, cada célula leva em consideração os estados dos seu vizinhos. Contudo, a presença de ruído pode corromper a informação obtida de um vizinho com probabilidade $\eta/2$, isto é, com uma certa chance o estado reconhecido não é o real estado dessa célula vizinha. O ruído não altera o estado da célula, mas somente como ela é lida pelos seus vizinhos. Consequentemente, por causa do ruído, a mesma célula pode enviar informações diferentes para seus vizinhos. É importante notar que o ruído não afeta a informação que a célula tem do seu próprio estado.

Outro importante componente presente em sistemas reais é o ruído. Nós o incorporamos no modelo dentro do processo de comunicação entre as células. O efeito do ruído é de corromper a informação obtida dos vizinhos de acordo com uma dada probabilidade de $\eta/2$ – com $\eta$ variando de 0 a 1. Cada estado para ser atualizado deve construir um padrão de *input*, resultado da leitura dos estados dos seus vizinhos. Nesse ponto entra o ruído, pois apesar do estado de um dado vizinho ser por exemplo 1, esse estado pode ser lido erradamente como 0 com uma probabilidade de $\eta/2$, como mostra a Fig. 4.4. Um estudo detalhado da regra da maioria aplicada em um autômato com ruído e topologia de "mundo-pequeno" foi realizado por Moreira *et al.* [**?**]. Nesse trabalho os autores mostram que a regra da maioria é uma estratégia heurística simples capaz de resolver o problema da maioria.

O redirecionamento das conexões na rede de mundo-pequeno ajuda o sistema a atingir a configuração de consenso, mas o ruído se mostrou um ingrediente fundamental. Na ausência do ruído, o sistema evolui para grupos de células que tendem a manter a sua opinião, enquanto que na presença do ruído esses grupos desaparecem, fazendo com que a regra da maioria tenha uma eficiência alta, como mostrado nos diagramas de configuração da Fig. 4.5. Deve-se notar que o ruído não muda o estado da célula, de fato, ele apenas altera a informação recebida pelo seu vizinho.

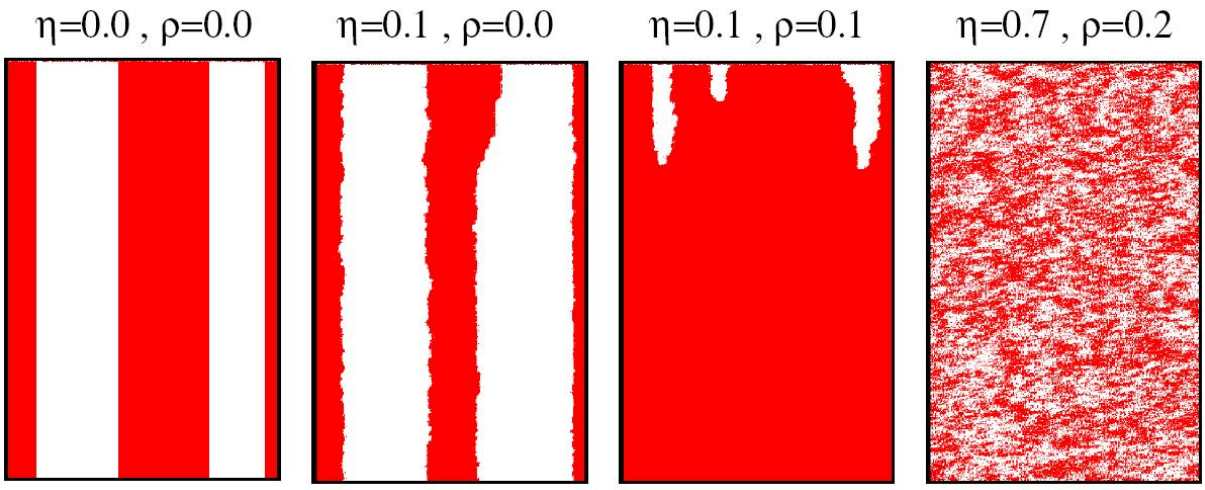



Figura 4.5: Efeito do ruído e da probabilidade de redirecionamento na dinâmica temporal de um sistema com $N = 299$ e $k = 11$, operando de acordo com a regra da maioria. As cores vermelhas e brancas representam as duas possibilidades de estado. Com $\eta = 0.0$ e $\rho = 0.0$ o sistema para em um configuração de domínios estáveis e nunca atinge o consenso. Quando o ruído e a probabilidade de redirecionamento são aumentados o sistema é capaz de convergir para o consenso de acordo com o estado de maioria inicial. Nós observamos que as conexões aleatórias na rede ajudam o sistema a atingir o estado final correto, mas o ruído é um ingrediente fundamental para quebrar as paredes dos domínios.

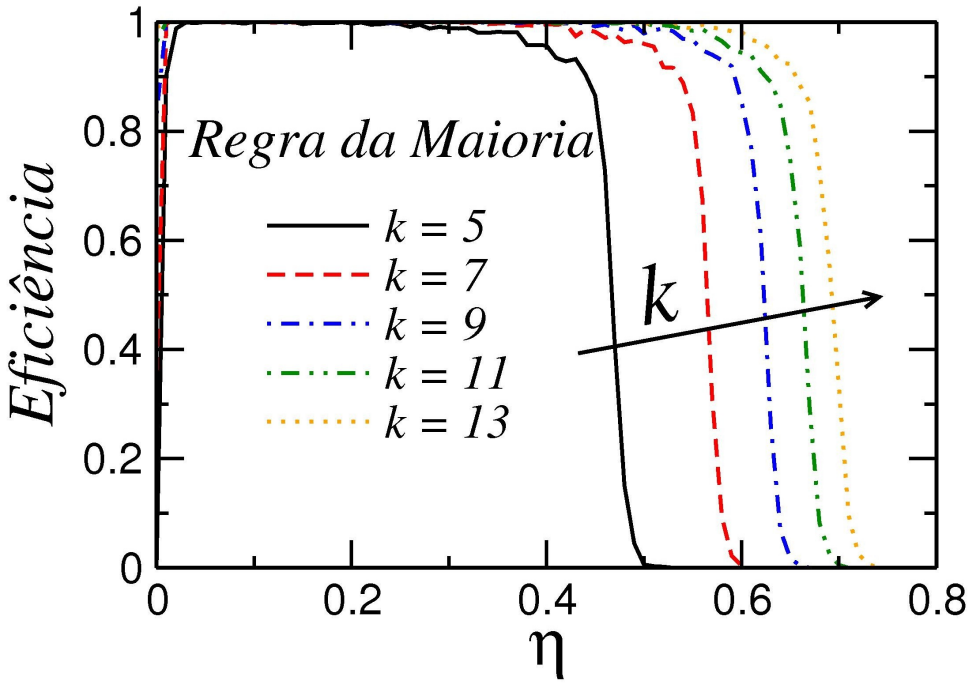



Figura 4.6: Eficiência da regra da maioria contra a intensidade do ruído para um sistema com $N = 299$ e $\rho = 0.3$. Como mostrado no gráfico, quanto maior o número de vizinhos, mais robusta a regra da maioria é ao efeito do ruído.

Um importante aspecto do nosso modelo é que temos dois elementos que introduzem desordem, o ruído ($\eta$) e a probabilidade de redirecionamento ($\rho$). O crescimento exagerado do nível de ruído tende a fazer com que as regras tenham uma queda na eficiência (*fitness*). Como esperado, uma regra com mais vizinhos (i.e., $k$ maior), como tem acesso a mais informação, tenderá a ser mais robusta à ação do ruído, como confirmado pela simulação com a regra da maioria na Fig. 4.6. A eficiência de uma regra $\phi$, $E\phi(\rho, \eta, N)$, é uma função do nível de ruído $\eta$, da probabilidade de redirecionamento $\rho$, e do tamanho do sistema $N$. A *classificação por densidade* é atingida caso todos os estados convirjam para o estado de maioria inicial, sendo que nenhum crédito parcial é dado. A eficiência é portanto a porcentagem do total de condições iniciais que uma dada regra $\phi$ classifica corretamente.

Para cada regra $\phi$ testamos 100 condições iniciais para um dado conjunto de parâmetros $(\rho, \eta, N)$. Com isso estimamos $E\phi(\rho, \eta, N)$ como sendo a fração de classificações corretas dentro de $2N$ passos de tempo. Para considerar um conjunto não tendencioso de condições iniciais, metade das condições iniciais têm 60% de 0's e a outra metade tem maioria 1, com inicialmente 60% de 1's. Devemos fazer desse modo, pois do contrário evoluiríamos regras triviais, onde qualquer *input* estaria associado a um *output* igual a 0, por exemplo. Essas regras teriam um *fitness* de 100% caso as condições iniciais fossem só de maioria 0. Estamos interessado em regras capazes de realizar computação, e portanto isso deve independer de que tipo de maioria inicial tivermos (0 ou 1).

A introdução do ruído resulta em um grau de aleatoriedade na dinâmica do autômato, por conseguinte o completo consenso pode nunca surgir. Iremos supor que a classificação está completa quando qualquer desvio da classificação correta é causado por essa flutuação. A cada passo de tempo devemos "desligar" o ruído e observar qual a configuração atingida. Caso seja a configuração de consenso, então paramos a dinâmica daquela condição inicial e a registramos como uma classificação correta.

Para $k = 7$ o tamanho da regra é $2^7 = 128$ *bits*, logo há $2^{128}$ regras possíveis – número muito grande para se testar por exaustão. Vemos que o número de possíveis regras cresce super-exponencialmente com $k$, tornando impraticável o teste de todas. Enquanto algumas regras eficientes já são conhecidas [**?**], é necessário entender em que direção as regras podem evoluir, e que tipos de regras podem evoluir "naturalmente". Como já explicamos detalhadamente, podemos usar o Algoritmo Genético como método de busca e estudo da evolução dessas regras. Desta forma, utilizamos o algoritmo explicado na seção 3.4 com uma população de 100 regras. A cada geração medimos o *fitness* de cada

uma dessas regras e selecionamos as 20 melhores para fazerem parte da próxima geração e gerar as 80 restantes através de crossover e mutação. A probabilidade de mutação $m$ varia dependendo do tamanho da regra, no caso inicial de $k = 5$ usamos $m = 0.05$, mas a medida que aumentamos $k$ devemos diminuir esse valor de $m$. Isso ocorre pois futuramente usaremos um algoritmo de promoção em que as regras com valor grande de $k$ serão construídas com base nas regras evoluídas com $k$ pequeno, sendo assim, essas novas regras terão *bits* que devem ser preservados. Caso a taxa de mutação for muito grande, simultaneamente partes desejadas e indesejadas das regras serão alteradas atrasando o processo de evolução.

Finalmente podemos aplicar o Algoritmo Genético com o objetivo de analisar a função de eficiência com o passar das gerações. Partindo de uma população inicial escolhida aleatoriamente, primeiramente nos restringimos à evolução de regras com um pequeno número de vizinhos, $k = 5$. Para esse valor de $k$ dentro da população inicial, algumas das regras já têm fitness 0.5, o que significa que elas classificam corretamente metade das condições iniciais, mas de maneira tendenciosa, sem realizar qualquer tipo de computação mais complexa. O algoritmo trabalha de modo que as regras evoluam para regras que acertem ambas as condições iniciais, as iniciadas com maioria 1 e as com maioria 0. Na Fig. 4.7 vemos o processo de evolução através da variação da eficiência da melhor regra em cada geração. Como explicado inicialmente a melhor regra tem eficiência 0.5, mas essa eficiência cresce rapidamente e com poucas gerações encontramos regras com *fitness* próximos de 1.0. É interessante, também, comparar as regras à medida que evoluem com a regra da maioria. Para isso, medimos a similaridade entre elas através da distância de Hamming entre as duas regras, ou seja, fração de *bits* idênticos a elas. A similaridade inicialmente cresce à medida que selecionamos regras mais eficientes, contudo, quando o máximo de eficiência é atingido, não há mais "pressão" de evolução e a similaridade atinge um valor estacionário, com similaridade de aproximadamente 90%. No *inset* da Fig. 4.7 mostramos qual o número de gerações necessárias para se encontrar regras eficientes (tipicamente com fitness $> 0.9$) em função do valor de $k$. Como vemos, o número de gerações necessárias cresce super-exponencialmente com $k$, consequentemente não podemos encontrar regras eficientes para o caso de $k = 9$ sem um grande custo computacional, que em média está acima de 5000 gerações.

Antes de tentarmos partir para uma busca de soluções com valores grandes de $k$, devemos analisar mais detalhadamente as soluções encontradas para $k = 5$ e 7 em comparação com a regra da maioria. Primeiramente, tomando a regra da maioria, podemos modificá-la aleatoriamente produzindo novas regras. Ao plotarmos a eficiência dessas regras à medida

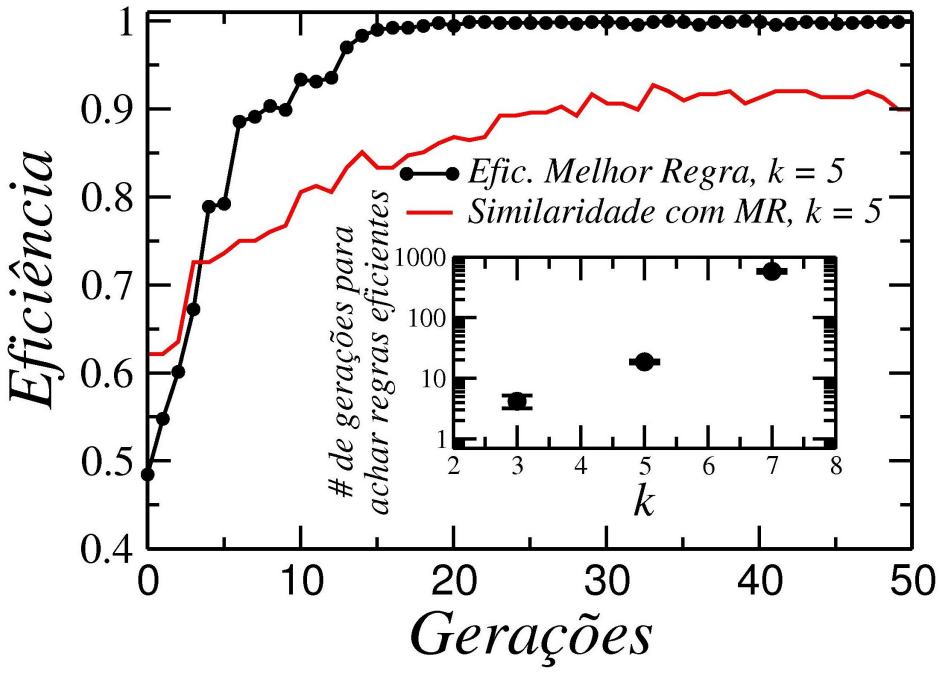



Figura 4.7: Eficiência contra o número de gerações para o caso de $k = 5$, $N = 99$, $\rho = 0.3$ e $\eta = 0.1$. Vemos que o Algoritmo Genético é capaz de encontrar, sobre essas condições, regras com *fitness* de aproximadamente 1.0 para um pequeno número de gerações. Também mostramos a similaridade da melhor regra, em cada passo de tempo, com a regra da maioria. Vale notar que a similaridade da melhor regra com a regra da maioria atinge um patamar de 90%. No *inset* mostramos que a média do número de gerações necessárias para se encontrar regras eficientes cresce muito rapidamente com o número de vizinhos $k$.

que aumentamos a distância para a regra da maioria, vemos que a densidade de regras eficientes é relativamente baixa mesmo para regras muito similares a da maioria, como ilustra a Fig. 4.8. Vemos um número grande de regras com *fitness* perto de 0.5 e muitas outras com *fitness* 0.0 à medida que aumentamos a distância. Isso indica que nem todos os *bits* têm o mesmo valor, de modo que nem todas as regras 90%, ou principalmente 80%, similares à regra da maioria terão grande eficiência. Por esse motivo, é necessário entender que características uma dada regra deve ter para ser eficiente.

Para descobrir mais sobre a classe de regras eficientes, tomamos 100 realizações do algoritmo, obtendo 100 regras eficientes com $k = 7$. A representação final de cada regra é salva. Comparamos então a similaridade com a regra da maioria, só que neste caso medimos a similaridade para cada *bit*. Como para $k = 7$ temos 128 *bits*, contamos para cada um deles quantas vezes o seu valor (0 ou 1) é igual à regra da maioria. Alguns *bits* são iguais aos da regra da maioria em praticamente todas as regras evoluídas, enquanto que outros apresentam somente 60% de similaridade. Esse fato pode ser interpretado como evidência de que alguns *bits* são necessários para produzir uma regra eficiente,

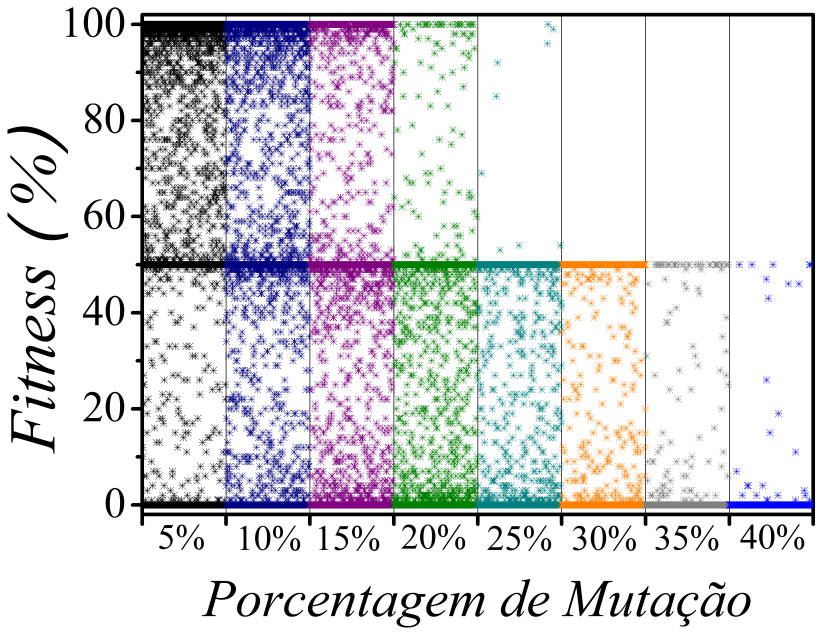



Figura 4.8: *Fitness* da regra da maioria degradada. Realizamos mutações aleatórias sobre a regra da maioria por $x\%$ em que $N = 399$, $k = 7$, $\eta = 0.2$ e $\rho = 0.2$. Para cada porcentagem de mutação foram geradas aleatoriamentes 10000 regras.

enquanto outros têm efeitos menores. Para descobrir o que alguns *bits* têm de especial, analisamos a que tipo de *input* este *output* está relacionado. Observamos que a frequência da similaridade dos *bits* está relacionada com o peso de 0's e 1's no padrão de *input*. O que podemos ver na Fig. 4.9 é que se a razão entre 1's e 0's é 7/0 ou 6/1, o *bit* correspondente deve ser igual ao da regra da maioria aproximadamente 100% das vezes. Enquanto que *bits* com razão 3/4 ou 4/3 são idênticos a regra da maioria apenas em 65% das vezes. A medida que crescemos a razão entre 0's e 1's, ou entre 1's e 0's, a concordância com a regra da maioria deve ser maior. De fato, a medida que essa razão cresce significa que a célula está exposta a um padrão de *input* de maioria explícita, ou todos tem a mesma opinião ou apenas uma célula tem opinião diferente, não dando margem a equívocos. Se esses *bits* não seguirem a regra da maioria o sistema não poderá atingir o consenso. Mas talvez a informação mais importante dada por esse gráfico seja que o mais relevante é a razão entre 0´s e 1´s no padrão de *input* não sendo necessário analisar o padrão exato do *input*. Essa característica irá nos permitir descobrir o funcionamento de regras mais complicadas mais a frente.

Como regras com mais vizinhos (i.e. com mais informação) são mais robustas a ação do ruído, deve haver alguma vantagem evolutiva em incluir mais vizinhos. Contudo, mais informação claramente aumenta a complexidade das estratégias, tornando muito

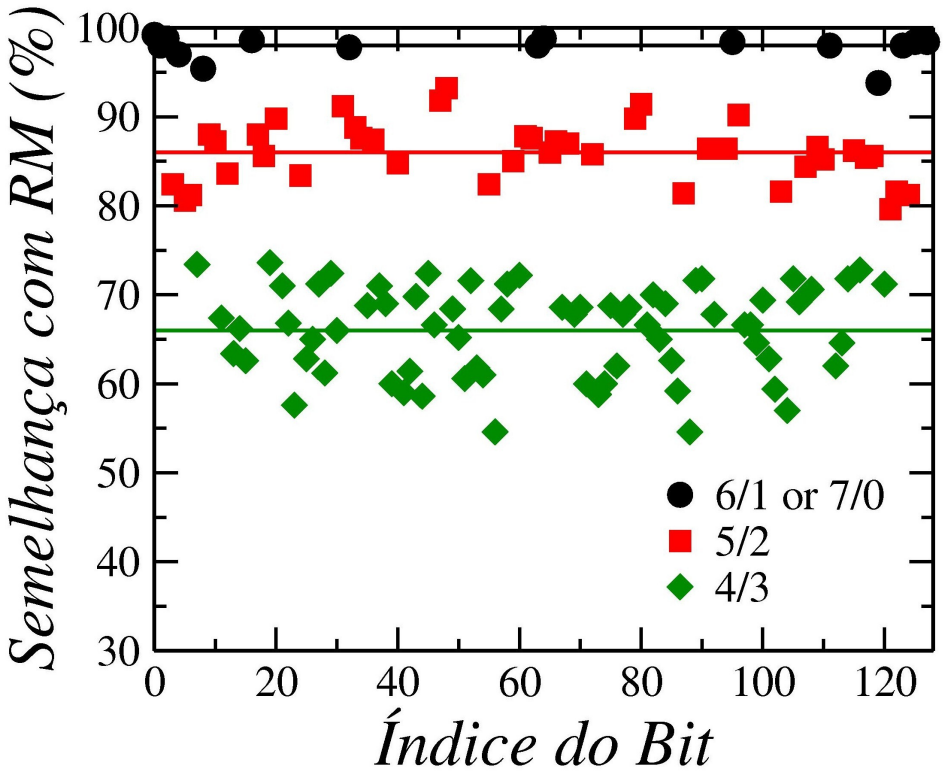



Figura 4.9: Porcentagem de regras evoluídas contendo o *bit* específico igual ao da regra da maioria (RM). A importância dos *bits* em uma regra de 128 *bits*, em que $k = 7$, $N = 99$, $\eta = 0.1$ e $\rho = 0.5$, está relacionada com a razão dos *bits* no padrão de *input*. O gráfico mostra que para os *bits* com razões 7/0 e 6/1 é essencial ser idêntico a regra da maioria para ser eficiente.

difícil para uma busca iniciada aleatoriamente pelo Algoritmo Genético achar uma regra eficiente. Como é muito difícil achar regras eficientes para $k \geq 7$, isso sugere que a evolução deve construir soluções complexas baseadas em soluções para problemas simples. É de se esperar que a Evolução Natural não tenha começado com genomas contendo milhões de genes, mas sim com a evolução de pequenos genomas que eventualmente devem ter aumentado até chegarem ao tamanho dos atuais.

Com o propósito de superar esse obstáculo propomos um algoritmo de promoção para as regras, usando o ruído como "pressão" evolutiva. Para explicar o algoritmo podemos imaginar uma situação de laboratório. Imagine que temos uma população de bactérias que são inicialmente sensíveis a um dado veneno. Queremos fazer com que elas fiquem imunes a esse veneno, caso as coloquemos junto a uma solução de concentração alta desse veneno provavelmente vamos matar todas, mas se ao invés disso adicionamos o veneno de modo gradual, o diluindo bastante em água, algumas morrerão, mas selecionaremos as mais fortes. Após o período de adaptação, repetimos o processo, adicionando mais veneno,

porém agora em uma concentração ligeiramente maior. Ao repetirmos esse processo, até atingir altas concentrações do veneno, ao final obteremos bactérias imunes.

Baseado na explicação acima, propomos um algoritmo em que o ruído, fazendo o papel do veneno, tem a sua intensidade aumentada ao passar do tempo criando uma "pressão" evolutiva. Começamos com regras simples, com $k = 5$, e um pequeno nível de ruído, $\eta = 0.1$. Depois de poucas gerações, regras de *fitness* próximo de 1.0 aparecem. A cada momento que uma regra eficiente aparecer, nós aumentamos o ruído a uma taxa de 0.01. Esse processo deve se repetir até que, após 50 gerações, se o algoritmo não é mais capaz de achar um regra eficiente (i.e. *fitness* acima de 0.98), então promovemos as regras para incluir mais dois vizinhos. A promoção deve ser de modo tal que a nova regra compute o mesmo que a regra anterior, para garantir que não estamos introduzindo novas informações de modo artificial. De fato, podemos construir uma regra de tamanho $k = 7$ que tenha exatamente o mesmo comportamento que uma regra de tamanho $k = 5$. Para construir essa nova regra introduzimos uma ambiguidade, fazemos com que a introdução dos dois novos vizinhos seja arbitrariamente irrelevante. Para isso basta que o *output* seja sempre igual ao da regra $k = 5$, não importando qual combinação dos estados dos dois novos vizinhos. Uma descrição esquemática do método de promoção é mostrada na Fig. 4.10.

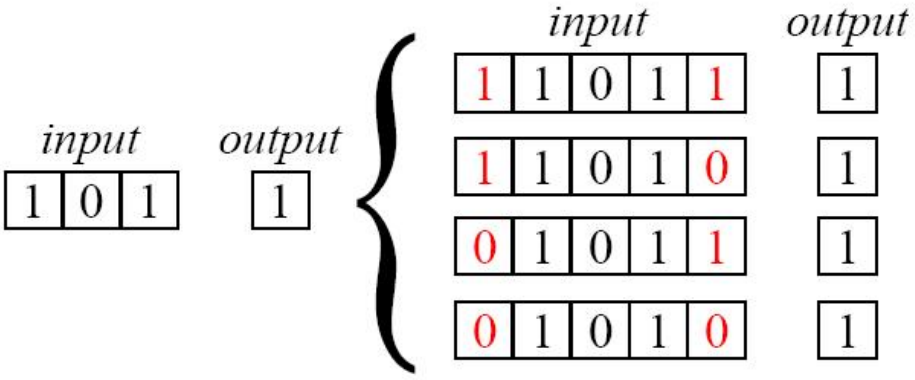



Figura 4.10: Para promover a regra de $k$ para $k+2$, incluímos 2 vizinhos extras mantendo o mesmo *output* para qualquer combinação dos estados desses dois novos vizinhos. Na *esquerda* vemos um possível *input* e seu *output* para uma regra $k = 3$. Na *direita* aparecem os respectivos possíveis *inputs* com dois novos vizinhos para um regra $k = 5$. A nova regra irá computar essencialmente o mesmo que a regra antiga, isto é, os novos *inputs* não afetam o comportamento da regra. A mutação e o crossover do Algoritmo Genético podem mudar esses *outputs* fazendo com que os novos vizinhos sejam eventualmente relevantes para a computação da regra evoluída.

Apesar das regras promovidas terem inicialmente o mesmo *fitness* que as de $k$ menor, o aumento do nível de ruído atua como uma "pressão" evolutiva para que as regras in-

corporem essa nova informação, do contrário elas "morrem". Por isso, é importante que o aumento do ruído seja gradual, do contrário todas as regras vão a níveis muito baixos de *fitness* estagnando o processo de evolução. A mutação e o crossover envolvidos no Algoritmo Genético vão gradualmente mudando as regras promovidas, modificando os novos *bits*, de modo que a regra agora possa identificar os novos vizinhos e utilizar dessa informação extra para melhorar a sua eficiência em ambientes cada vez mais ruidosos. Na Fig. 4.11 mostramos que esse método mostrou ser bastante eficiente ao evoluir regras até $k = 11$ em condições de grande ruído, $\eta = 0.7$, algo antes impossível sem o auxílio do algoritmo de promoção.

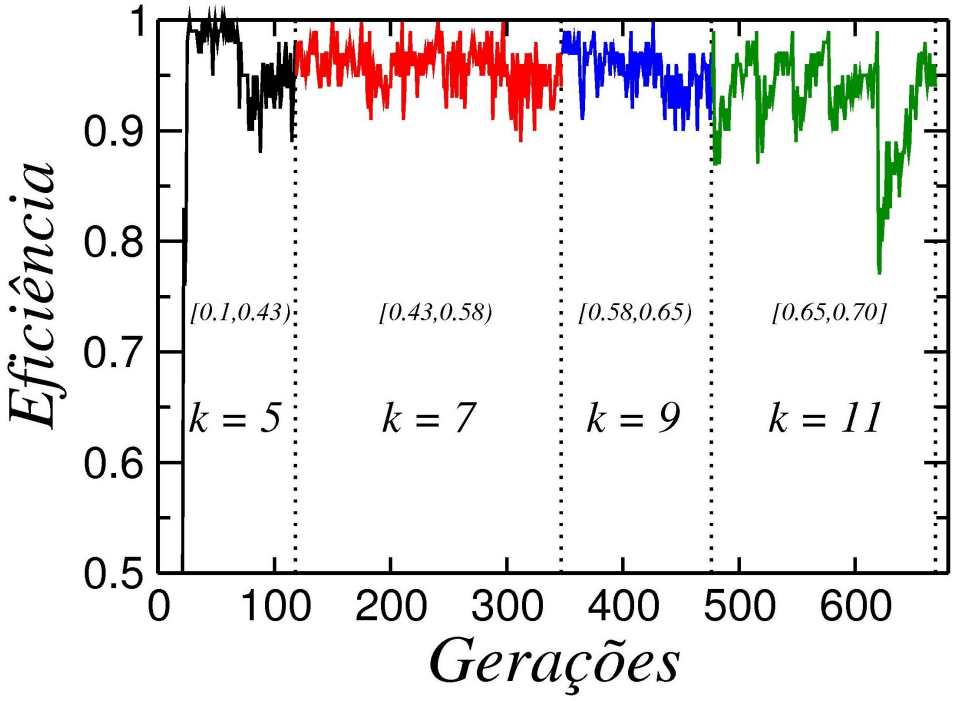



Figura 4.11: Eficiência em função do número de gerações. Começamos com uma população de regras $k = 5$ e um valor baixo do ruído, $\eta = 0.1$. Quando uma regra evolui e atinge a eficiência de 0.98, a intensidade do ruído é aumentada em 0.01. Se após aumentar o ruído nenhuma regra evoluir para atingir novamente uma eficiência de 0.98 nas próximas 50 gerações, nós promovemos as regras da população para incluir 2 novos vizinhos. A regra promovida é no início essencialmente idêntica à antiga, com os vizinhos extras agindo como *inputs* mudos, isto é, a informação extra não é usada pela regra. O Algoritmo Genético, através da mutação e do crossover, deve evoluir regras eficientes que consigam utilizar esses novos *inputs*. Através desse método, podemos obter regras eficientes até $k = 11$ para altos valores do ruído, por exemplo, $\eta = 0.7$.

Após encontrarmos regras eficientes para valores mais altos de $k$, devemos estudar as propriedades dessas regras, mais especificamente a razão pela qual tais regras são eficientes, mesmo para esses níveis de ruído. Para realizar esse estudo salvamos a melhor regra para cada $k$ e analisamos a robustez (*fitness* versus ruído) dela comparando com a regra

da maioria para o mesmo $k$. Notamos na Fig. 4.12 que, ao contrário do que esperávamos inicialmente, as regras evoluídas são mais robustas ao efeito do ruído do que a regra da maioria. Para entendermos este comportamento, devemos analisar detalhadamente a diferença entre as regras evoluídas e a regra da maioria.

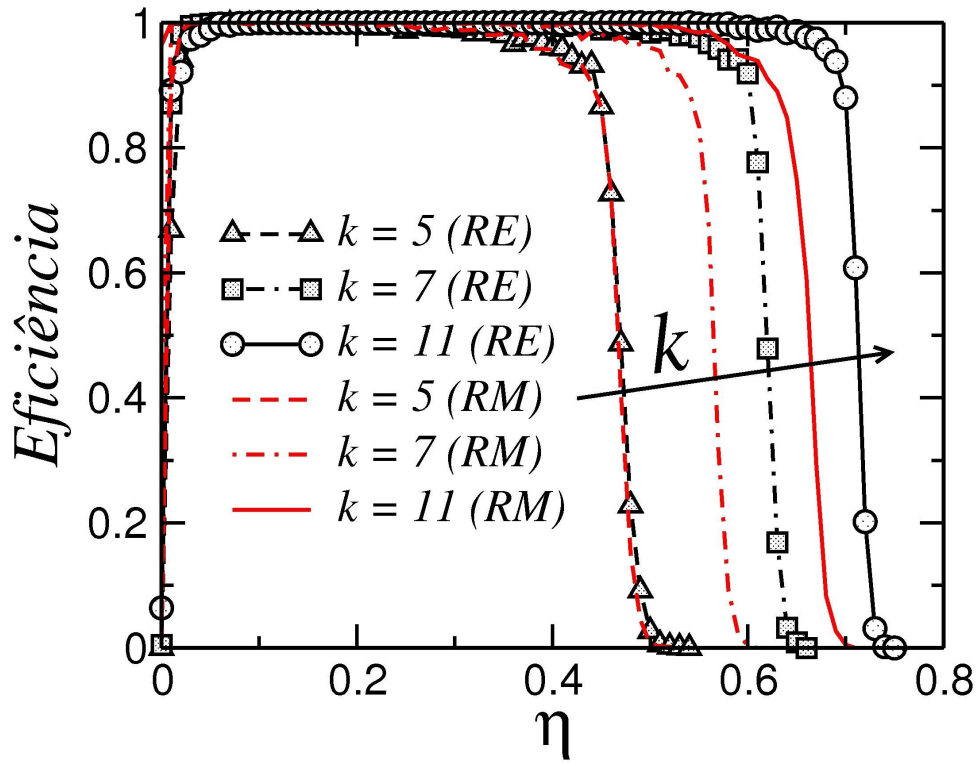



Figura 4.12: Gráfico com a eficiência das regras evoluídas (RE) comparada com a eficiência da regra da maioria (RM) contra a intensidade do ruído. Podemos ver que, para o mesmo número de vizinhos, as regras evoluídas são mais robustas ao efeito do ruído que a regra da maioria correspondente.

Lembremos do resultado anterior de que podemos ignorar o padrão de *input* exato para todos os *bits* e devemos nos focar na densidade de 1's ou 0's nesse padrão. Fazer isso nada mais é que construir um gráfico de limiar, como explicado no início desse capítulo. A Fig. 4.13 mostra qual é a probabilidade de mudar para o estado 0 ou 1, dependendo do número de 1's no padrão de *input*. Nesta figura mostramos a média no ruído e nos valores dos *outputs* para todos os padrões de *inputs* possíveis com um número de 1's fixado. Na Fig. 4.13 podemos ver que, à medida que vamos para situações extremas, por exemplo, o caso em que temos uma unanimidade de 1's, a curva da regra da maioria fica abaixo da curva da regra evoluída. Isso mostra que a regra evoluída identifica mais vezes de modo correto essa unanimidade – atribuindo ao próximo estado o valor 1 – do que a regra da maioria, porém isso ainda não explica o motivo da robustez dessas regras evoluídas. No *inset* da Fig. 4.13 vemos a mesma curva de limiar para a regra evoluída com $k = 11$, em

comparação com a regra da maioria, mas neste caso sem ruído. Vemos que a mudança de opinião na regra evoluída ocorre de maneira mais suave e não descontínua como no caso da regra da maioria. Essa suavidade mostra um comportamento de persistência que entendemos como um indício do funcionamento das regras evoluídas.

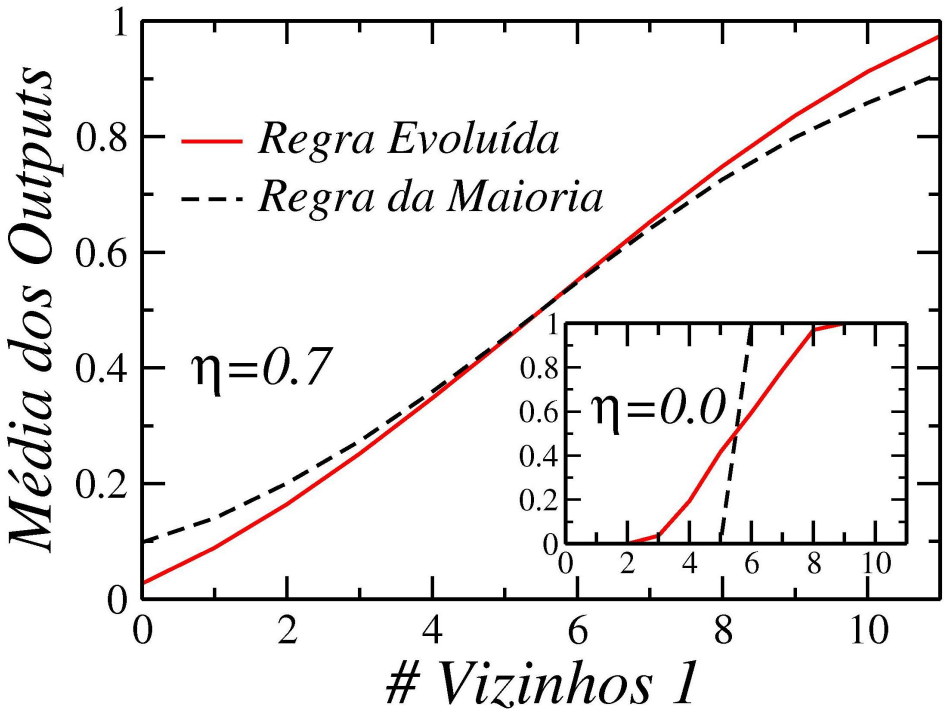



Figura 4.13: Curva de limiar, ou média dos *outputs* para a regra evoluída e regra da maioria com $k = 11$. Medida da probabilidade de mudar para o estado 1, em dois regimes de ruído. Para $\eta = 0.7$, vemos que a regra evoluída erra menos que a regra da maioria quando observamos as situações de maioria extrema. No *inset* vemos o caso em que $\eta = 0.0$. Nele podemos observar o comportamento conservador da regra evoluída, mudando de opinião em 100% dos casos apenas quando houver uma razão de 9/2 no *input*.

A eficiência das regras evoluídas para grandes níveis de ruído pode ser explicada em termos do maior peso dado por essas regras à informação envolvendo a própria célula (não influenciada pelo ruído) do que à informação vinda dos seus vizinhos, que naturalmente pode ter sido corrompida pelo ruído. Na Fig. 4.14 construímos o mesmo gráfico de limiar, mas agora o dividimos em duas contribuições. Uma das curvas (pretas preenchidas) mostra qual é a probabilidade de uma célula no estado 0 mudar para o estado 1, em função do número de vizinhos com estados 1. O segundo tipo de curva (vermelhas tracejadas) mostra o mesmo, mas agora para uma célula de estado inicial 1. Quando comparamos a regra evoluída (curvas com símbolos) com a regra da maioria (curvas sem símbolos), para um ambiente sem ruído, vemos claramente que, na regra evoluída, para se mudar de estado são

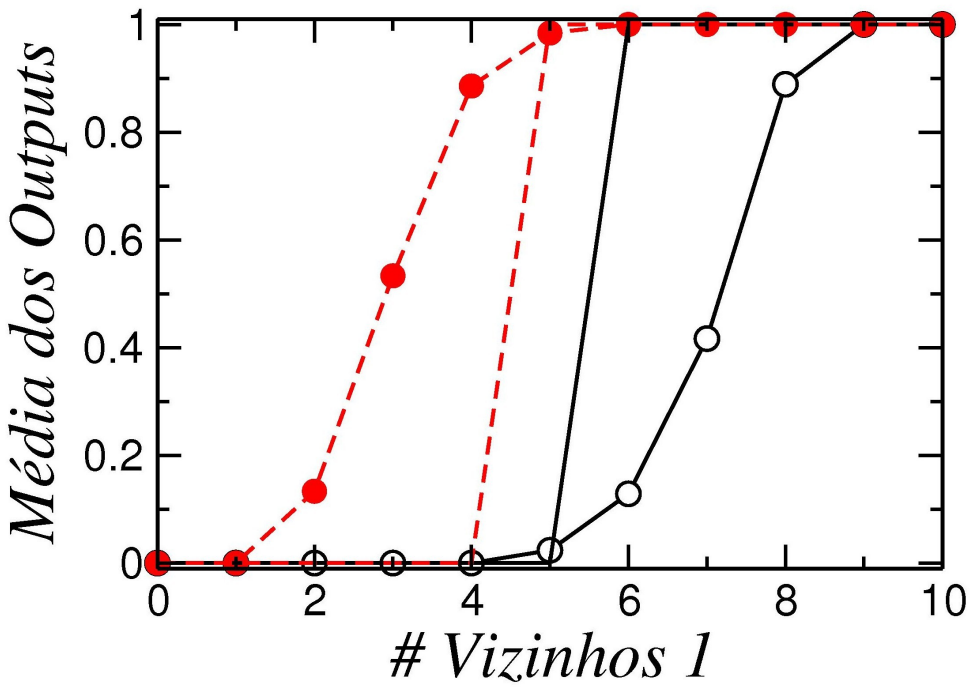



Figura 4.14: Curva de limiar, ou médias dos *outputs*, para a regra evoluída. As curvas vermelhas tracejadas correspondem a célula no estado 1, e as pretas continuas a célula no estado 0. Quando seis dos vizinhos estão de acordo com o estado 1, enquanto que a célula tem estado 0, seguindo a regra da maioria, temos uma mudança de estado. Já as células seguindo a regra evoluída nessa mesma situação podem continuar no mesmo estado, mesmo não concordando com a maioria dos vizinhos. O resultado indica que, em um ambiente ruidoso, é benéfico ser conservador e dar maior peso à sua opinião do que a dos vizinhos.

necessárias muitas "opiniões" opostas, enquanto que, na regra da maioria, basta apenas um estado a mais do que a metade do número de vizinhos para que isto ocorra. Como consequência, as células que seguem as regras evoluídas tendem a manter seus próprios estados, a não ser que uma grande maioria não concorde com eles. Essa *persistência*, ou comportamento *conservador*, pode ser visualizada pelos padrões verticais observados no diagrama de configurações para a regra evoluída com $k = 11$ e ruído $\eta = 0.68$, como mostra a Fig. 4.15. Nesse nível a regra evoluída ainda é eficiente, ao contrário da regra da maioria.

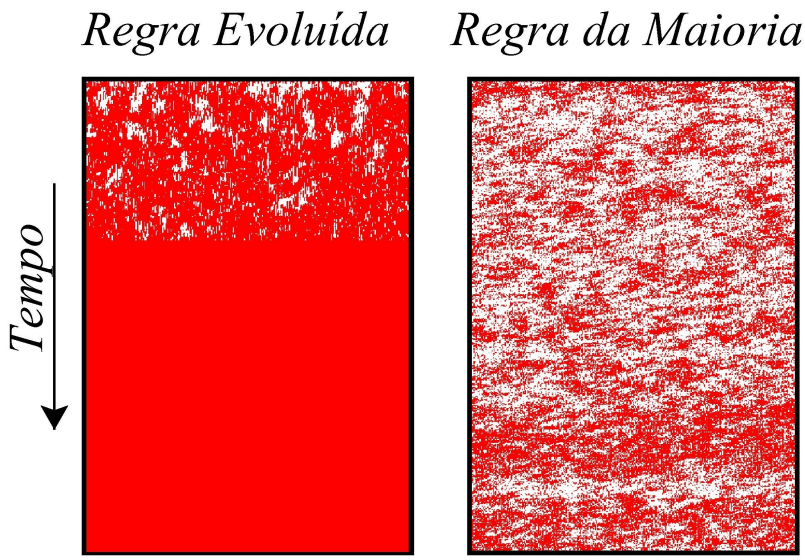



Figura 4.15: Comportamento da regra evoluída e da maioria com $k = 11$ e $\eta = 0.68$. Para esse nível de ruído, a regra da maioria não consegue mais atingir o consenso, enquanto que a regra evoluída consegue. O padrão vertical presente no diagrama da regra evoluída indica que as células tendem a manter seus estados por mais tempo, caracterizando o que chamamos de comportamento *conservador*.

# 5 Conclusões e Pespectivas

Os resultados apresentados nesse trabalho demonstram que o ruído e a topologia afetam criticamente a performance das estratégias para atingir coordenação global e processamento de informação. Vimos que uma simples regra heurística como a regra da maioria é eficiente nesses ambientes. Estudamos até que ponto a regra da maioria ainda é eficiente à medida que aumentamos o ruído, caracterizando a sua robustez. Como redes sociais e naturais são caracterizadas pela presença de ruído nas transmissões e topologias complexas, nosso estudo propicia uma explicação para o uso já observado de tais regras heurísticas por animais e humanos [**?**, **?**]. Assim como no trabalho [**?**], vemos que as regras de decisão não podem ser avaliadas de modo isolado do ambiente. O sucesso na coordenação global e comportamento coletivo depende tanto da regra quanto do ambiente em que ela está sendo aplicada. Nesse sentido, a regra da maioria é "ecologicamente eficiente", sendo adequada para simular sistemas que se assemelham ao mundo real. Isso leva a possibilidade de que as redes e as regras "ecologicamente eficientes" tenham coevoluído ao passar do tempo.

Apesar do modelo de Evolução de Autômatos Celulares ser simples, ele exibe um comportamento complexo como o encontrado em muitos sistemas naturais, o que faz com que o seu estudo possa servir de base para uma melhor compreensão de sistemas descentralizados. Os primeiros resultados descritos são um indício do valor evolutivo de regras do tipo-maioria para se atingir o consenso. Um mecanismo tipo-maioria pode ser importante no entendimento de muitos sistemas, desde coordenação em neurônios [**?**] até aplicações em economia [**?**] e processos de decisão em sociologia e psicologia [**?**].

A dificuldade em evoluir regras eficientes para $k > 5$, nos sugere que a evolução deve construir soluções complexas baseadas naquelas de problemas mais simples. A Evolução Natural certamente não começou com genomas contendo milhares de genes. Ela deve ter-se iniciado com genomas menores que eventualmente aumentaram de tamanho permitindo assim uma maior complexidade, mas sem perder as informações obtidas anteriormente. Baseado nisso, criamos um método de promoção no tamanho de regras que nos permite

obter regras eficientes em regimes de ruído em que regras pequenas não "sobrevivem". Inspirados em experimentos de laboratório, aplicamos um aumento gradual no ruído, obrigando as regras a se adaptarem às novas condições "ambientais". Esse aumento do ruído gera uma "pressão" evolutiva, obrigando que as regras promovidas levem em conta os novos vizinhos para que possam sobreviver. A regra de promoção, junto com o aumento apenas gradual do ruído, nos permitiu obter regras eficientes de tamanho até $k = 11$ para grandes regimes de ruído. Acreditamos ser possível, com esse algoritmo, evoluir regras com $k$ acima do que já encontramos. Para isso seria necessário diminuir ainda mais a taxa com a qual aumentamos o ruído à medida que aumentamos $k$. Como temos um limite superior para o ruído, $\eta = 1.0$, esse refinamento é necessário, pois a robustez das novas regras será cada vez mais próxima da robustez das regras anteriores.

O entendimento do funcionamento das regras obtidas pelo Algoritmo Genético constituiu um dos maiores desafios do trabalho junto à correta idealização do algoritmo de promoção. De posse apenas da representação vetorial da regra evoluída com $k = 11$ não podemos identificar, à primeira vista, qual é a causa da sua eficiência. Uma regra com $k = 11$, à primeira vista, nada mais é do que uma grande sequência 0's e 1's, onde a identificação de qualquer padrão é proibitiva. Para uma análise detalhada precisariamos reduzir o número de informações presentes no vetor da regra. Isso só foi possível quando levamos em conta a possibilidade da construção da curva de limiar, baseado no fato de que a informação mais importante era a frequência de 1's ou 0's (probabilidade de escolha) para uma dada densidade de 1's no padrão de *input*. A visualização da curva de limiar nos permitiu identificar o comportamento peculiar das regras evoluídas, mais robustas que a regra da maioria. Verificamos que um comportamento de persistência, ou conservador, evolui "naturalmente" como uma forma de defesa contra a informação corrompida pelo ruído.

O comportamento conservador existe no sentido de que a célula do autômato necessita de uma grande maioria de estados opostos para mudar de estado, ao contrário do que ocorre na *regra da maioria*. Esse comportamento é justificado porque, em um ambiente de informações não confiáveis, é melhor manter a sua opinião por mais tempo, levando à uma convergência para o estado de consenso mais lenta, porém mais robusta. Esse comportamento teórico nos parece concordar com o comportamento encontrado por Ash no seu experimento de conformidade [**?**]. As pessoas pareciam alterar a sua opinião quando expostas a uma maioria contrária, porém era necessário que houvesse apenas um apoio para que elas tivessem coragem de manter as suas convicções. Podemos interpretar o experimento considerando que essas pessoas estavam em um ambiente de grande ruído,

pois devido ao fato das respostas corretas serem muito óbvias, as respostas dos atores eram claramente duvidosas. Portanto, podemos imaginar que havia uma competição entre dois comportamentos, um de conformidade e outro de persistência, compatível com as regras evoluídas em nosso modelo. Vale ressaltar que, ao mesmo tempo que obtemos regras capazes de explicar este fenômeno, o nosso próprio método de busca de solução, o Algoritmo Genético, também constitui parte do estudo. A relevância dos nossos resultados se deve ao fato de que eles podem ser obtidos por meio de uma dinâmica estocástica provavelmente presente na própria evolução da natureza.

# *Referências Bibliográficas*


[1] Young, H. P. (1996) J. Econ. Perspect. 10, 105–122.

[2] Boyd, R. & Richerson, P. J. (1995) Ethol. Sociobiol. 16, 125–143.

[3] Heyes, C. M. & Selten, R., eds. (1996) Social Learning in Animals: The Roots of Culture (Academic, New York).

[4] Bikhchandani, S., Hirshleifer, D. & Welch, I. (1998) J. Econ. Perspect. 12, 151–170.

[5] Gigerenzer, G. & Selten, R., eds. (2001) Bounded Rationality: The Adaptative Toolbox (MIT Press, Cambridge, MA).

[6] Amaral, L. A. N. & Otino, J. M. (2004) Eur. Phys. J. B 38, 147–162.

[7] Amaral, L. A. N. & Otino, J. M. (2004) Chem. Eng. Sci. 58, 1653–1666.

[8] Mitchell, M. (2008). Complexity: A Guided Tour. New York: Oxford University Press.

[9] Dawking, R., (1986). The Blind Watchmaker. New York: W. W. Norton & Company.

[10] Nussenzveig, H. M. (Org.) . Complexidade e Caos. 1. ed. Rio de Janeiro: Editora UFRJ/COPEA, 1999. v. 01. 276 p.

[11] Gács, Péter; Kurdyumov, G. L.; Levin, L. A. (1978). "One dimensional uniform arrays that wash out finite islands"(em Russo). Problemy Peredachi Informatsii 14: 92–98.

[12] "We will never understand complex system unless we map out and understand the networks behind them". Albert-László Barabási, (2011). Network Science: Introduction January 10, curso online encontrado em: http://barabasilab.neu.edu/courses/phys5116/

[13] A. E. Eiben and J.E. Smith (2003). Introduction to Evolutionary Computing, Springer.

[14] L.J. Fogel, A.J. Owens, and M.J. Walsh (1966) Artificial Intelligence through Simulated Evolution, John Wiley, NY.

[15] Ingo Rechenberg (1973), Evolutionsstrategie - Optimierung technischer Systeme nach Prinzipien der biologischen Evolution, Stuttgart: Frommann-Holzboog.

[16] Schwefel, H.-P. (1965) Kybernetische evolution als strategie der experimentellen forschung in der stroemungstechnik, Dipl-Ing. Thesis, Technical Univ. Berlin.

[17] Schwefel, H.-P. (1995) Evolution and Optimum Seeking Wiley, New York.

[18] J. H. Holland (1973). Genetic algorithms and optimal allocation of trials. SIAM J. of Computing, 2pp. 88 – 105.

[19] Holland, J. H. (1975). Adaptation in Natural and Artificial Systems. University of Michigan Press. (Second edition: MIT Press, 1992.)

[20] Box, G. E. P. 1957. Evolutionary operation: A method for increasing industrial productivity. Journal of the Royal Statistical Society C 6, no. 2: 81–101.

[21] Friedman, G. J. 1959. Digital simulation of an evolutionary process. General Systems Yearbook 4: 171–184.

[22] Bledsoe, W. W. 1961. The use of biological concepts in the analytical study of systems. Paper presented at ORSA–TIMS National Meeting, San Francisco.

[23] Bremermann, H. J. 1962. Optimization through evolution and recombination. In M. C. Yovits, G. T. Jacobi, and G. D. Goldstein, eds., Self–Organizing Systems. Spartan Books.

[24] Reed, J., Toombs, R., and Barricelli, N. A. 1967. Simulation of biological evolution and machine learning. Journal of Theoretical Biology 17: 319–342.

[25] Darwin, Charles (1859). On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life (1st ed.). London: John Murray.

[26] Wright, Sewall (1932). The roles of mutation, inbreeding, crossbreeding and selection in evolution. Proc. 6th Int. Cong. Genet 1: 356–366.

[27] Brodie, E. D., III. 1992. Correlational selection for color pattern and antipredator behavior in the garter snake Thamnophis ordinoides. Evolution 46:1284-1298.

[28] Richard Dawkins. Climbing Mount Improbable. New York: Norton, 1996.

[29] Wright, Sewall (1982). The Shifting Balance Theory and Macroevolution. Annual Review of Genetics 16: 1–19.

[30] Garrett Hardin, The Tragedy of the Commons, Science, Vol. 162, No. 3859 (December 13, 1968), pp. 1243-1248.

[31] Watson J, Crick F (1953). Molecular structure of nucleic acids; a structure for deoxyribose nucleic acid. Nature 171 (4356): 737–8.

[32] A. J. Keane and S. M. Brown (1996), "The design of a satellite boom with enhanced vibration performance using genetic algorithm techniques," pp. 107-113 in Proceedings of the Conference on Adaptive Computing in Engineering Design and Control 96, ed. I. C. Parmee,P.E.D.C., ISBN 0 905227 61 1, Plymouth.

[33] A.-L. Barabási, R. Albert (1999). Emergence of scaling in random networks. Science 286, 509–512.

[34] M. E. J. Newman and M. Girvan (2004). Finding and evaluating community structure in networks, Phys. Rev. E 69, 026113.

[35] Das, R., Mitchell, M., and Crutchfield, J. P. 1994. A genetic algorithm discovers particle-based computation in cellular automata. In Y. Davidor, H. -P. Schwefel, and R. Männer, eds., Parallel Problem Solving from Nature–PPSN III. Springer-Verlag (Lecture Notes in Computer Science, volume 866).

[36] Mitchell, M., Hraber, P. T., and Crutchfield, J. P. 1993. Revisiting the edge of chaos: Evolving cellular automata to perform computations. Complex Systems 7, 89–130.

[37] Schultz TR (2000). "In search of ant ancestors". Proceedings of the National Academy of Sciences 97 (26): 14028–14029.

[38] Ulam, S., 1974, "Some ideas and prospects in biomathematics," Ann. Rev. Bio., 255.

[39] von Neumann, J., 1963, "The general and logical theory of automata," in J. von Neumann, Collected Works, edited by A. H. Taub, 5, 288.

[40] von Neumann, J., 1966, Theory of Self-Reproducing Automata, edited by A. W. Burks (University of Illinois, Urbana).

[41] Wiener, N.; Rosenblueth, A. (1946). "The mathematical formulation of the problem of conduction of impulses in a network of connected excitable elements, specifically in cardiac muscle". Arch. Inst. Cardiol. México 16: 205.

[42] Hordijk, W., Manderick B., The Usefulness of Recombination, Proceedings of the Third European Conference on Advances in Artificial Life, p.908-919, June 04-06, 1995.

[43] Konrad Zuse, 1969. Rechnender Raum. Braunschweig: Friedrich Vieweg & Sohn. 70 pp.; Uma tradução para o inglês feita pelo *MIT Technical Translation* pode ser acessada em http://cag.dat.demokritos.gr/Backpages/zuserechnenderraum.pdf

[44] Gardner, M., 1971, "Mathematical games," Sci. Amer. 224, February, 112; March, 106; April, 114.

[45] Gardner, M., 1972, "Mathematical games," Sci. Amer. 226, January, 104.

[46] Conway, J. H., 1970, unpublished.

[47] Wolfram, S., 1982a, "Cellular automata as simple self-organizing systems," Caltech preprint CALT-68-938 (submitted to Nature).

[48] S. Wolfram, Statistical mechanics of cellular automata, Rev. Modern Phys., 55 (1983) 601.

[49] S. Wolfram, "Cellular automata as models of complexity", Nature 311 (1984) 419–424.

[50] S. Wolfram, "Origins of randomness in physical systems", Physical Review Letters 55 (1985) 449–452.

[51] Wolfram, Stephen, A New Kind of Science. Wolfram Media, Inc., May 14, 2002.

[52] W. Daniel Hillis (1989-02). "Richard Feynman and The Connection Machine". Physics Today. Retrieved 3 November 2006.

[53] O artigo de W. Daniel Hillis pode ser encontrado junto com outros, em homenagem a Feynman, compilados no livro *O melhor de Feynman*, Ciência Aberta gradiva, 1994.

[54] Fuks, Henryk; Boccara, Nino (1998). "Generalized deterministic traffic rules". Journal of Modern Physics C 9(1): 1–12.

[55] Belitsky, Vladimir; Ferrari, Pablo A. (1995). "Ballistic annihilation and deterministic surface growth". Journal of Statistical Physics 80 (3–4): 517–543.

[56] Cook, Matthew (2004) "Universality in Elementary Cellular Automata", Complex Systems 15: 1-40.

[57] Crutchfield, J. P and Mitchell, M. The evolution of emergent computation.(1995). Proc. Natl. Acad. Sci. U.S.A. 92, 10742.

[58] Moreira, A., Mathur, A., Diermeir, D., and Amaral, A.N. (2004). Efficient System-wide Coordination in Noisy Environments. Proc. Natl. Acad. Sci. U.S.A. 101 (33), 12085–12090.

[59] Matthew Champion, "Re: How many atoms make up the universe?", 1998.

[60] Mitchell, M., Crutchfield, J. P., and Hraber, P. T. (1994). Evolving cellular automata to perform computations: Mechanisms and impediments. Physica D 75, 361.

[61] Mitchell, M. (1996). An Introduction to Genetic Algorithms. Cambridge, MA: MIT Press.

[62] Hanson, J. E., and Crutchfield, J. P. (1992). The attractor-basin portrait of cellular automaton. Journal of Statistical Physics 66, no. 5/6: 1415–1462.

[63] Crutchfield, J. P., and Hanson, J. E. (1993). Turbulent pattern bases for cellular automata. Physica D 69:279–301.

[64] Asch, S. E. (1955). Opinions and social pressure. Scientific American, 193, 31-35.

[65] Watts, Duncan (2003). Six Degrees: The Science of a Connected Age. W. W. Norton & Company.

[66] Granovetter, M. (1978). "Threshold Models of Collective Behavior". American Journal of Sociology 83 (6): 1420.

[67] Milgram, S., Bickman, L., Berkowitz, L. (1969). Note on the drawing power of crowds of different size. Journal of Personality and Social Psychology, Vol. 13, No. (2). (October 1969), pp. 79-82.

[68] John S. MacDonald and Leatrice D. MacDonald. (1962). Chain Migration, Ethnic Neighborhood Formation and Social Networks, The Milbank Memorial Fund Quarterly, 42, 82–97.

[69] Rogers, Everett M. 1979. Network analysis of the diffusion of innovations, pp. 137-164 in P. Holland and S. Leinhardt (eds.) Perspectives on social network research.

[70] Watts, D. J. and Strogatz, S. H. (1998) Nature 393,440–442.

[71] Watts, D. (1999) Small Worlds (Princeton University Press, Princeton, US).