



Paulo Sérgio Germano da Silva

Teste de Parâmetros da Busca Evolutiva por Autômatos  
Celulares no Problema da Paridade, com Ênfase nos  
Métodos de Avaliação das Regras

Dissertação apresentada ao  
Departamento de Pós-Graduação em  
Engenharia Elétrica da Universidade  
Presbiteriana Mackenzie para  
obtenção do Grau de Mestre

Orientador: Dr. Pedro Paulo Balbi de Oliveira

São Paulo

2007

# **Livros Grátis**

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

Paulo Sérgio Germano da Silva

Teste de Parâmetros da Busca Evolutiva por Autômatos  
Celulares no Problema da Paridade, com Ênfase nos  
Métodos de Avaliação das Regras

Dissertação apresentada ao  
Departamento de Pós-Graduação em  
Engenharia Elétrica da Universidade  
Presbiteriana Mackenzie para  
obtenção do Grau de Mestre

Aprovado em Setembro de 2007

**BANCA EXAMINADORA**

---

Dr. Pedro Paulo Balbi de Oliveira  
Universidade Presbiteriana Mackenzie

---

Dr. Luiz Henrique Alves Monteiro  
Universidade Presbiteriana Mackenzie

---

Dr. Stephan Stephany  
Instituto Nacional de Pesquisas Espaciais, INPE

*A Deus Pai que me dá forças e me  
guia. Aos meus pais Raymundo e  
Neusa, e à minha amada esposa  
Denise*

## **AGRADECIMENTOS**

Ao meu orientador Prof. Dr. Pedro Paulo, que me apresentou as fronteiras deste admirável mundo novo da pesquisa científica, pacientemente me orientando nos caminhos trilhados, sempre mostrando sua mão amiga nos momentos mais difíceis desta caminhada.

À minha amada esposa Denise que, com sua força, sabedoria, apoio e paciência me incentivou nos vários dias, e noites, necessários para o desenvolvimento deste trabalho, carregando sozinha o pesado fardo dos encargos familiares.

Aos meus amados pais, Raymundo e Neusa, que me concederam a sagrada dádiva da vida, e sempre foram a base de todas as minhas realizações.

Este trabalho conta com apoios financeiros concedidos a meu orientador pela Wolfram Research, através do *Mathematica Academic Grant* Ref 1149, Mackpesquisa, e pela FAPESP, através do projeto de pesquisa Proc. 2005/04696-3.

A todos meus profundos e sinceros agradecimentos.

## ÍNDICE

<b>Resumo</b> .....	10
<b>Capítulo I: Introdução</b> .....	14
1.1.    Objetivo.....	15
1.2.    Formulação do problema.....	15
1.3.    Metodologia aplicada.....	16
<b>Capítulo II: Conceitos</b> .....	18
2.1.    Autômatos.....	18
2.2.    Autômatos celulares.....	21
2.2.1.    Classificação.....	25
2.2.2.    Computação Implícita e Explícita a ACs.....	28
2.3.    O problema da classificação de densidade.....	29
2.4.    Computação evolutiva.....	31
<b>Capítulo III: Experimentos realizados</b> .....	39
3.1.    Parâmetros base.....	39
3.2.    Variação do número de gerações.....	42
3.3.    Distribuição quase-balanceada.....	42
3.4.    Distribuição quase-balanceada com avaliação parcial.....	43
3.5.    Configurações iniciais com distribuições generalizadas..	44
3.6.    Reticulado variável.....	47
3.6.1.    Reticulado variável após 30 gerações.....	47
3.6.2.    Reticulado variável desde a primeira geração.....	48
3.6.3.    Reticulados com comprimento ímpar.....	49
3.6.4.    Reticulados com comprimento par.....	50

3.6.5.	Reticulado com comprimento crescente por intervalo de gerações.....	51
3.7.	Um novo método de recombinação.....	52
3.7.1.	Distribuição binomial.....	55
3.7.2.	Distribuição quase-balanceada.....	56
3.7.3.	Reticulado com comprimento crescente por intervalo de gerações.....	57
3.8.	Novos parâmetros de busca.....	57
3.8.1.	Janelas progressivamente crescente .....	58
3.8.2.	Janelas de tamanho praticamente fixos, com sobreposição de comprimentos.....	59
3.9.	Pré-avaliação das CIs.....	60
<b>Capítulo IV: Conclusão.....</b>		<b>63</b>
<b>Capítulo V: Referências Bibliográficas.....</b>		<b>65</b>

## Índice de Figuras

Figura 2.1: Autômato com pilha.....	20
Figura 2.2: (a) Vizinhaça unidimensional; (b) Vizinhaça de Von Neumann; (c) Vizinhaça de Moore.....	22
Figura 2.3: (a) Representação da regra 210; (b) Aplicação da regra 210, 150 vezes.....	23-24
Figura 2.4: Condição de contorno periódico.....	24
Figura 2.5: Representações da regra 254 sobre condições iniciais (CIs) binárias aleatórias.....	25
Figura 2.6: Representações da regra 170 sobre condições iniciais binárias aleatórias.....	26
Figura 2.7: Representações da regra 30 sobre condições iniciais binárias aleatória.....	26
Figura 2.8: Representação da regra 110 aplicada durante 250 passos.....	28
Figura 2.9: (a) Condição inicial com quantidade de 1s superior a de 0s; (b) . Condição inicial com quantidade de 0s superior a de 1s.....	30
Figura 2.10: Demonstração de solução do problema da paridade conforme [Lee, Xu e Chau, 2001].....	32
Figura 2.11: Aplicação da regra 280363545044958142736483883671228394256 sobre um reticulado de comprimento Ímpar.....	34
Figura 2.12: Aplicação da regra 280363545044958142736483883671228394256 sobre um reticulado de comprimento Par.....	34
Figura 3.1: Exemplo de gráfico de CIs com distribuição uniforme.....	42

Figura 3.2: Distribuição quase balanceada com uma proporção de 90% de 1s. ....	43
Figura 3.3: Exemplo de gráfico de CIs com distribuição quase-balanceada e avaliação parcial.....	44
Figura 3.4: Gráfico de distribuição binomial com curvas generalizadas a direita e a esquerda.....	45
Figura 3.5: Aplicação de distribuições generalizadas.....	46
Figura 3.6: (a) CI com probabilidade de 1s próxima de 100% e .paridade ímpar (b) CI com probabilidade de 1s próxima de 50% e .paridade par.....	46
Figura 3.7: Evolução com reticulado variável após a 30ª geração.....	48
Figura 3.8: Evolução com reticulado variável a partir da primeira geração.....	49
Figura 3.9: Evolução com reticulado variável a partir da primeira geração, durante 300 gerações.....	49
Figura 3.10: Evolução com reticulados de comprimento ímpar.....	50
Figura 3.11: (a) Evolução para intervalos crescentes a cada 10 gerações; (b) Regressão do processo evolutivo gerada por variação de intervalos crescentes a cada 10 gerações.....	51-52
Figura 3.12: Representação da função de recombinação .....	54
Figura 3.13: Teste do novo modelo de recombinação na DCT.....	54
Figura 3.14: Novo modelo de recombinação com distribuição binomial.....	55
Figura 3.15: Novo modelo de recombinação com distribuição binomial, por 300 gerações.....	56
Figura 3.16: Novo modelo de recombinação com distribuição quase-balanceada.	56
Figura 3.17: Novo modelo de recombinação com reticulado com comprimento crescente por intervalo de gerações.....	57

Figura 3.18: Janelas progressivamente crescentes.....	59
Figura 3.19: Janelas de tamanhos praticamente fixos, com sobreposição de comprimento.....	59

## Índice de Tabelas

Tabela 2.1: Regras de excelente nível de desempenho, para reticulados de comprimento ímpar relatadas em [Wolz e de Oliveira, 2007].....	31
Tabela 3.1: Representação da forma de fracionamento dos cromossomos para $j = 1$ .....	53

## Resumo

A obtenção de soluções de problemas através da computação implícita a regras de autômatos celulares permanece um assunto desafiador, pelo pouco que ainda se conhece a respeito. Entre vários problemas abordados na literatura, um que tem despertado interesse é o problema da paridade, em que a meta é determinar a paridade da quantidade de 1s de uma cadeia binária fornecida como configuração inicial a um autômato celular unidimensional binário. Apesar de clássico, ainda não se tem registro de uma regra capaz de resolver o problema e, de fato, nem se sabe se essa solução existe. Nesse contexto, apresentam-se aqui os resultados experimentais de buscas evolutivas realizadas para encontrar boas regras de autômatos celulares unidimensionais binários de raio 3, no problema da paridade. O foco principal da pesquisa foi verificar o efeito de se variar as estratégias de geração das amostras de configurações iniciais de teste, o que se traduz em diferentes formas de avaliação das regras candidatas. Tal enfoque expande os trabalhos de [de Oliveira e Vaiano, 2005], caminhando na direção de melhor compreender e reproduzir alguns dos resultados obtidos em [Wolz e de Oliveira, 2007], em que regras de excelente desempenho foram achadas para reticulados de comprimento ímpar, bem como regras de bom desempenho, apesar de não robustas, para alguns comprimentos pares de reticulados. Apesar das várias abordagens testadas, nenhuma delas conduziu a boas regras no problema em questão, fazendo crer que a qualidade conseguida em [Wolz e de Oliveira, 2007] deve-se, portanto, aos outros conceitos lá empregados, distintos dos aqui avaliados.

**Palavras chaves**

Linguagens, Problema da Paridade, Autômatos Celulares, Computação Evolutiva, Algoritmo Genético, Computação não-convencional, Computação Emergente, NKS.

## **Abstract**

Problem solving through computation based on cellular automata rules remains a challenging subject, the scarcity of knowledge in the area. Among several problems reported in the literature, the parity problem has generated significant interest, its aim being to ascertain the parity of a quantity of 1s in a binary string, input as an initial configuration to a binary, one-dimensional, cellular automaton. Although a classic problem, no rule able to solve this problem has been found to date. Indeed, it remains uncertain as to whether such a solution exists. In this context, this work presents the results of evolutionary searches for sound rules of one-dimensional, cellular automata of radius 3, in the parity problem. The main focus of the research was to verify the effect of varying the strategies used to generate initial test configuration samples, which implied different ways of evaluating the candidate rules. Such an approach extends the work in [Oliveira e Vaiano, 2005] in the direction of better understanding as well as trying to reproduce some of the results in [Wolz e de Oliveira, 2007], where rules with excellent performance at odd-sized lattices were reported, as well as rules with good performance at even-sized lengths, even though the latter showed lack of robustness. In spite of testing a range of approaches, none led to good rules for the problem, leading to the conclusion that the excellent results obtained in [Wolz e de Oliveira, 2007] were due to the other concepts used therein, distinct from those evaluated in the present study.

**Key words**

Languages, Parity Problem, Cellular Automata, Evolutionary Computation, Genetic Algorithm, Unconventional Computation, Emergent Computation, NKS.

## Capítulo I: Introdução

A visão de autômato celular (AC) surgiu com John von Newman e seu contemporâneo Ulam, na busca de uma máquina capaz de se auto reproduzir [Von Neumann, 1966]. Caracterizado como um sistema discreto e dinâmico, os autômatos celulares (ACs) possuem propriedades computacionais implícitas e explícitas que lhe permitem amplo potencial de aplicações, embora ainda pouco explorados.

Dá-se o nome de computação implícita em autômatos celulares, ao resultado obtido pela aplicação de uma regra sobre uma condição inicial que, após um período discreto de aplicação da regra, gera uma configuração final que pode ser interpretada como a saída de uma operação computacional realizada sobre a condição inicial.

Uma característica marcante da computação implícita a regras de ACs é a sua imprevisibilidade, no sentido de que é extremamente difícil (e, em geral, impossível) saber a forma com que uma regra se comporta a partir de uma condição inicial aleatória, exigindo uma avaliação do resultado obtido após um certo número de passos de aplicação da regra.

Esta imprevisibilidade exige uma abordagem experimental junto ao conjunto de regras avaliadas, o que torna este campo de pesquisa extremamente intrigante e extenso, já que o número de regras e condições iniciais possíveis são enormes, constituindo um fecundo, e ainda pouco explorado, campo de trabalho.

Muitos trabalhos têm sido realizados na busca de soluções de problemas utilizando-se computação paralela implícita às regras dos autômatos celulares [Toffoli, 1987], entre os quais se destaca o problema de classificação da densidade (DCT), que consiste em identificar o bit predominante em uma cadeia binária de comprimento ímpar.

## 1.1. Objetivo

Motivado pelo desafio de obter soluções de problemas através da computação implícita a regras de autômatos celulares [Mitchell, 1996], visando suas potenciais aplicações, objetivou-se solucionar um problema matemático reconhecido que ainda encontra-se em aberto. Neste contexto buscam-se aqui regras que possam constituir soluções eficazes para o problema da paridade, o qual consiste em determinar a paridade da quantidade de 1s dos bits de uma cadeia binária, fornecida como condição inicial a um autômato celular unidimensional. Como é muito grande o número de regras e variáveis que influenciam no comportamento dos autômatos, gerando um grande número de possibilidades, utilizam-se aqui algoritmos de computação evolutiva [Mitchell, 1998] e [Jacob, 2001], a propósito muito utilizadas em problemas correlatos [de Oliveira, Bortot e Oliveira, 2006] e [Sipper 1998].

Como na DCT, onde várias regras têm sido encontradas com níveis de sucesso elevados, da ordem de 85% [de Oliveira, Bortot e Oliveira, 2006], a busca por boas regras que solucionem o problema da paridade sofreu uma série de avanços desde [Sipper, 1998] e [de Oliveira e Vaiano, 2005] como os relatados em [Wolz e de Oliveira, 2007] que obteve regras de excelente desempenho, em reticulados de comprimento ímpar.

## 1.2. Formulação do problema

Buscar uma solução para o problema da paridade através de computação implícita a regras de autômatos celulares representa a possibilidade de determinar a paridade de

uma cadeia binária por meio de um processamento puramente local, sem a utilização de uma pilha externa ao processo ou de mecanismos globais de processamento...

Nesse contexto buscam-se aqui regras de autômatos celulares (AC) que possam constituir soluções eficazes para o problema da paridade (PP), cuja formulação de interesse consiste na caracterização da paridade da quantidade de 1s em uma cadeia binária, segundo o que se segue:

- Se a quantidade de 1s é par, a aplicação da regra do AC sobre a cadeia deve converter todas as células para 1, ao final de um certo número de passos; e
- Caso contrário deve converter todas as células para 0, indicando que a quantidade é ímpar.

### 1.3. Metodologia aplicada

Devido ao grande número de regras que compõe a família de ACs unidimensionais binários de raio 3 (isto é, aqueles em que cada célula é influenciada por 3 vizinhos de cada lado), objetiva-se através de variações de um processo evolutivo, encontrar boas regras capazes de solucionar o problema da paridade. Neste intento, numa primeira etapa, foram realizadas buscas variando-se as distribuições da amostra de configurações iniciais de teste e o tamanho do reticulado, conforme linha de pesquisa descrita em [de Oliveira e Vaiano, 2005].

Numa segunda etapa, buscou-se variar o comprimento do reticulado e, posteriormente, alterar o processo evolutivo e o método de cruzamento utilizado no processo evolutivo; por fim, prosseguiu-se na linha de pesquisa descrita em [Wolz e de

Oliveira, 2007], seguindo os mesmos parâmetros de execução que permitiram obter as boas regras lá reportadas.

Utilizou-se como ferramenta de desenvolvimento para todos os experimentos aqui descrito o software Mathematica, sendo o algoritmo de busca evolutiva uma versão adaptada do sistema *Evolvica*, conforme descrito em [Jacob, 2001], alterada e otimizada para buscas evolutivas de autômatos celulares.

A seguir serão apresentados alguns fundamentos teóricos necessários para compreensão dos experimentos realizados e apresentados no Capítulo III, nele são abordados todos os aspectos envolvidos nestes, desde seus parâmetros base, as variações das condições iniciais utilizadas, o novo processo de recombinação aplicado ao processo evolutivo, variações dos parâmetros de busca e a aplicação de uma metodologia de pré-avaliação das condições iniciais.

Por fim é apresentada, como conclusão, uma análise crítica dos resultados obtidos e dos aspectos que envolveram este trabalho, sugerindo-se uma possível trajetória de continuidade desta pesquisa.

## Capítulo II: Referencial Teórico

### 2.1. Autômatos

*Autômato finito (determinístico)* - Um autômato finito [Hopcroft, Ullman e Motwani, 2005] é um dispositivo capaz de reconhecer linguagens do tipo 3 (linguagens regulares), sendo composto basicamente de três partes:

- *Fita de entrada*: É onde fica a cadeia a ser processada; ela é dividida em células, onde cada célula armazena um símbolo do alfabeto de entrada;
- *Unidade de controle*: Reflete o estado corrente da máquina. Contém uma cabeça de leitura (que pode ler uma célula da fita por vez) e movimenta-se exclusivamente numa única direção (por convenção, para a direita);
- *Função de transição*: Função que define o comportamento do autômato, dependendo da sua entrada e de seu estado atual.

O autômato finito não possui nenhum tipo de memória de trabalho, ou seja, ele não tem nenhum lugar onde armazenar conhecimento, a não ser em seus próprios estados.

Um autômato finito determinístico é representado pela quintupla:

$$M = (Q, \Sigma, \delta, q_0, F)$$

Os componentes possuem os seguintes significados:

$Q$  : conjunto finito de estados;

$\Sigma$  : alfabeto de entrada do autômato, composto por símbolos;

$\delta$  : função de transição, a qual é uma função parcial:  $\delta: Q \times \Sigma \rightarrow Q$

$q_0$  : elemento de  $Q$ , considerado o estado inicial;

$F$  : subconjunto de  $Q$  que constitui o conjunto de estados finais.

Um autômato finito determinístico [Sipser, 1997] sempre pára ao processar qualquer entrada, pois não há a possibilidade de um loop infinito, visto que a cadeia de entrada é finita e um novo símbolo tem sempre de ser lido à direita, a cada aplicação da função de transição.

Ao parar, o autômato finito pode se encontrar em um de dois estados: aceitação da cadeia de entrada ou sua rejeição. As condições de parada são:

- fim da cadeia de entrada e a cadeia foi reconhecida.
- fim da cadeia de entrada e a cadeia não foi reconhecida.
- a função transição não consegue tratar a condição atual do autômato (entrada + estado atual) e o autômato não reconhece a cadeia.

**Autômato finito não-determinístico** - O não determinismo de um autômato é definido da seguinte forma: dada uma condição qualquer do autômato, o seu próximo estado é na verdade um conjunto com um ou mais estados [Papadimitriou, 2000].

Um autômato finito não-determinístico é representado pela quintupla:

$$M = (Q, \Sigma, \delta, q_0, F)$$

Os componentes possuem os seguintes significados:

Q : conjunto finito de estados;

$\Sigma$  : alfabeto de entrada do autômato, composto por símbolos;

$\delta$ : função de transição, a qual é uma função parcial:  $\delta: Q \times \Sigma \rightarrow 2^Q$

$q_0$ : elemento de Q, considerado o estado inicial;

F : subconjunto de Q que constitui o conjunto de estados finais.

A classe dos autômatos finitos determinísticos é equivalente à classe dos autômatos finitos não-determinísticos. Isso quer dizer que eles têm o mesmo poder de reconhecimento gramatical: o não-determinismo de um autômato finito não

aumenta a classe de linguagens que o autômato pode reconhecer, visto que existe um método de conversão de um autômato finito não-determinístico para um autômato finito determinístico (adicionando novos estados e novas transições).

**Autômato de pilha (PDA)** - Da mesma maneira que o autômato finito é associado às linguagens regulares, ao autômato com pilha são associadas as linguagens livres de contexto, ou seja, um autômato com pilha é um dispositivo capaz de reconhecer linguagens do tipo 2.

A pilha é uma estrutura de memória de leitura e gravação e é dividida em células, armazenando, em cada uma, um símbolo do alfabeto da pilha.

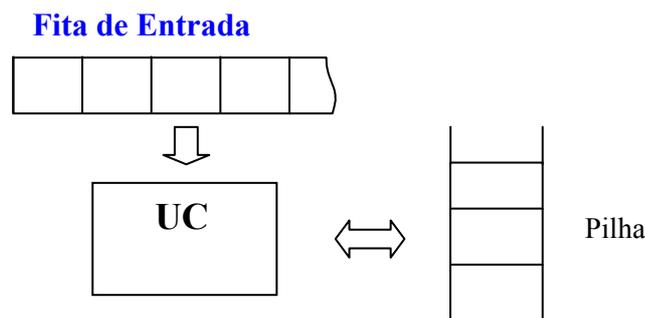


Figura 2.1: Autômato de pilha.

Esse autômato é composto basicamente de quatro partes:

- Fita de entrada: idêntica à do autômato finito;
- Pilha: memória auxiliar de leitura e gravação;
- Unidade de controle: reflete o estado corrente da máquina. Contém uma cabeça de leitura da fita e uma cabeça de leitura/escrita na pilha;
- Função de transição: comanda a leitura da fita, leitura e gravação da pilha e define o estado do autômato.

Um autômato de pilha é representado pela septupla:

$$M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$$

Os componentes possuem os seguintes significados:

$Q$  : conjunto finito de estados;

$\Sigma$  : alfabeto de entrada do autômato, composto por símbolos;

$\Gamma$  : alfabeto da pilha;

$\delta$ : função de transição, a qual é uma função parcial:

$$\delta: Q \times (\Sigma \cup \{ \varepsilon \}) \times \Gamma \rightarrow 2^{(Q \times \Gamma^*)} \text{ (não determinístico)}$$

$q_0$ : elemento de  $Q$ , considerado o estado inicial;

$Z_0$ : símbolo de início. Inicialmente, a pilha do PDA consiste de uma instância desse símbolo e em nada mais.

$F$  : subconjunto de  $Q$  que constitui o conjunto de estados finais.

A pilha é independente da fita de entrada, e é potencialmente infinita. Como toda pilha, é uma estrutura de dados do tipo LIFO (*last-in, first-out*), ou seja, o último a entrar é o primeiro a sair. A base da pilha é fixa e o seu topo é variável e define a posição do último símbolo gravado. As operações de leitura e escrita são sempre executadas na célula-topo da pilha, e o autômato sempre inicia o seu funcionamento com a pilha apresentando um símbolo inicial.

## 2.2. Autômatos celulares

Autômatos Celulares são modelos matemáticos formados por um reticulado regular, cujas posições (células) possuem características idênticas, onde são aplicadas regras de transição pré-definidas que alteram os valores (estados) discretos que elas podem assumir [Wolfram, 2002].

ACs evoluem em passos de tempo discreto, regidos por uma série de transições de estado (i.e., que forma sua regra de transição), as quais são aplicadas sobre o estado da célula em questão, de acordo com o próprio e o de suas células adjacentes, segundo uma vizinhança adequadamente definida.

No caso de um reticulado unidimensional são consideradas células vizinhas imediatas à célula em questão para um raio de amplitude  $r = 1$ . Assim, para um AC binário de  $r = 1$  tem-se uma vizinhança de tamanho  $1+2r = 3$ , como descrito na Figura 2.3 (a).

A vizinhança de uma célula pode ser definida como sendo todas as células de um reticulado que influenciam de forma local na transição de um estado. Os dois conceitos principais de vizinhança bidimensionais são ilustrados na Figura 2.2, porém outras configurações com  $n$  dimensões podem ser geradas.

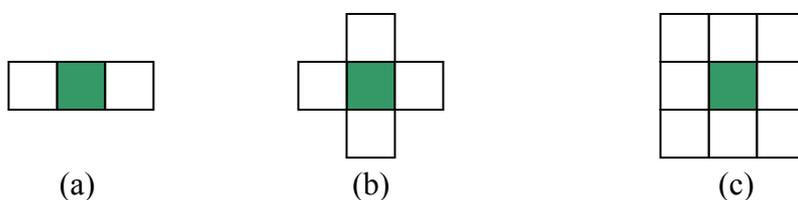


Figura 2.2: (a) Vizinhança unidimensional; (b) Vizinhança de Von Neumann; (c) Vizinhança de Moore.

Assumindo  $k$  como sendo o número de estados por célula, existem  $k^{1+2r}$  estados possíveis dessa família de ACs. No caso de regras binárias unidimensionais de raio 1, tem-se a família chamada de ACs elementares [Wolfram, 2002], constituída por  $2^8=256$  regras, as quais podem ser representadas pelos números de 0 a 255, de acordo com seus bits de saída, conforme ilustra a Figura 2.3, para a regra 210 (em binário 11010010). Tal

representação lexicográfica é utilizada de fato para qualquer outra família, inclusive no presente trabalho.

Uma regra simples para a evolução no tempo dos valores das células é

$$a^{(t+1)}_i = a^{(t)}_{i-1} + a^{(t)}_{i+1} \text{ mod } 2 ,$$

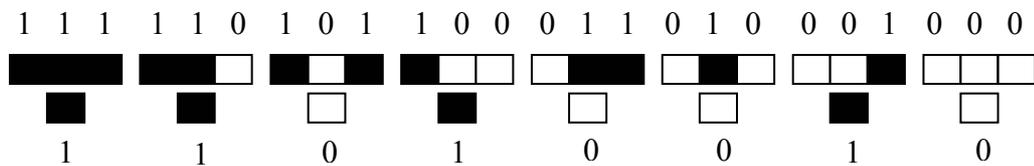
onde:

$i$  : posição da célula;

$t$  : tempo em valores discretos;

$a$  : valor da célula;

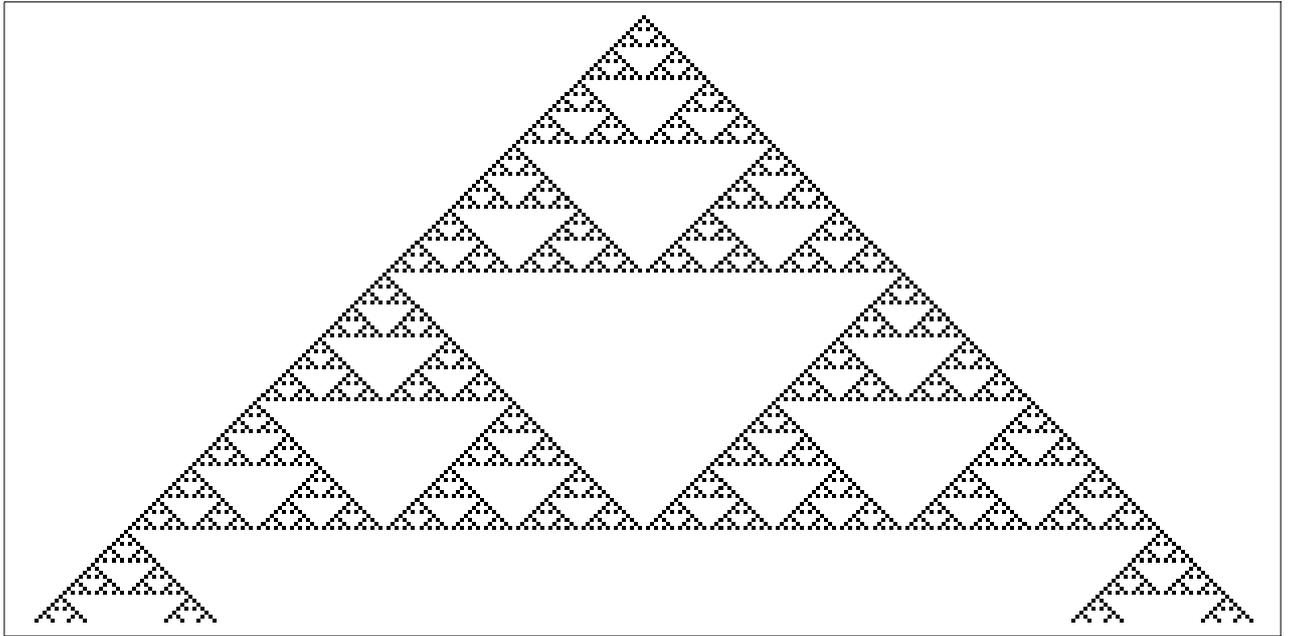
A seguir apresenta-se uma regra e sua aplicação durante um número fixo de passos, para um reticulado com configuração inicial de células formada representada pela cadeia ...0001000....



$$11010010_2 = 210$$

(a)

Reticulado inicial formado pela cadeia ...0001000...



(b)

Figura 2.3 (a) Representação da regra 210; (b) Aplicação da regra 210, 150 vezes.

A aplicação das regras de um AC normalmente está associada a uma condição de contorno periódica, onde as células iniciais do reticulado são associada às últimas, isto permite formar a vizinhança das primeiras e das últimas células do reticulado, sendo que, para raio 1 a última célula será a vizinha a esquerda da primeira célula, o mesmo ocorre com a última célula que assume como vizinha a direita a primeira célula do reticulado, fechando um círculo. Os número de células que deverão compor a condição de contorno periódica varia conforme de acordo com o raio a ser adotado, veja ilustração abaixo.

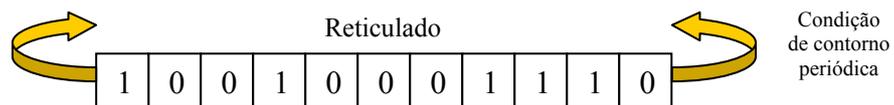


Figura 2.4: Condição de contorno periódica.

O conceito original de autômato celular surgiu com Von Neumann [1966], em sua busca por conexões entre a biologia e a teoria dos autômatos. Em seus estudos, predominava a idéia do fenômeno biológico da auto-reprodução, e a questão que ele procura responder era: “*Que tipo de organização lógica é suficiente para um autômato ser capaz de reproduzir a si próprio?*” Outros cientistas, contemporâneos de Von Neumann, desenvolveram trabalhos relacionados, tais como [Arbib, 1969] e [Myhill, 1964].

### 2.2.1. Classificação

Wolfram [Wolfram, 1984; Wolfram, 2003] visualizou uma classificação dos ACs elementares em quatro famílias:

**Classe 1:** É caracterizada por um comportamento simples, onde as informações da condição inicial convergem rapidamente todas para o mesmo valor. No caso de um AC binário estas regras convergem todas para 1 ou 0. A regra 254 é um bom exemplo desta classe, conforme mostra a Figura 2.5, em que a regra é aplicada sobre 5 reticulados de 30 células.

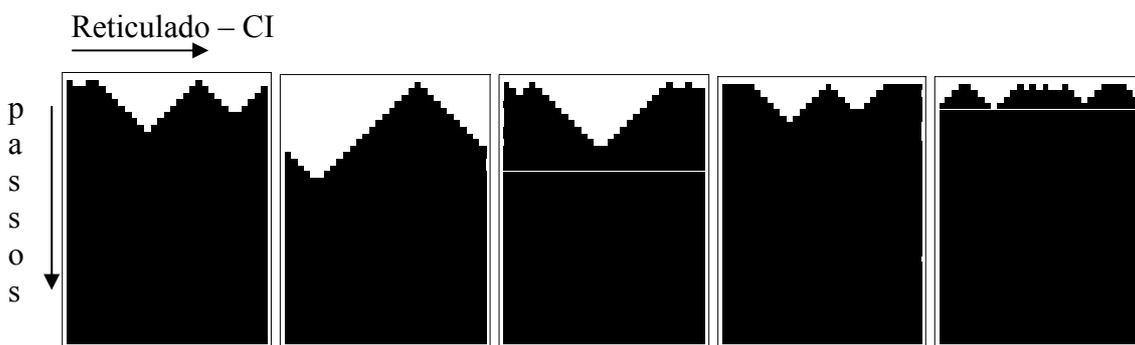


Figura 2.5: Representações da regra 254 sobre condições iniciais (CIs) binárias aleatórias.

**Classe 2:** É também caracterizada por um comportamento simples, porém com variações periódicas bem definidas. Nesta classe algum padrão do reticulado inicial é

mantido na configuração final, mesmo após um certo número de passos, porém, seu posicionamento é alterado. A regra 170 é um bom exemplo desta classe, conforme a Figura 2.6, em que a regra é aplicada sobre 5 reticulados de 30 células.

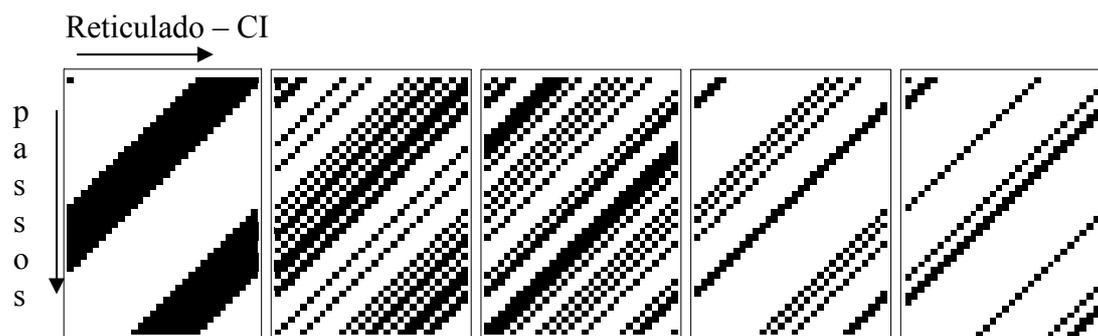


Figura 2.6: Representações da regra 170 sobre condições iniciais binárias aleatórias.

**Classe 3:** Nesta classe o AC possui um comportamento aleatório e muitas vezes caótico, podendo ocorrer repetições de partes dos padrões iniciais mesmo após vários passos de execução da regra. A Figura 2.7 ilustra essa classe, onde mostra-se a regra 30 aplicada sobre 5 reticulados de 30 células.

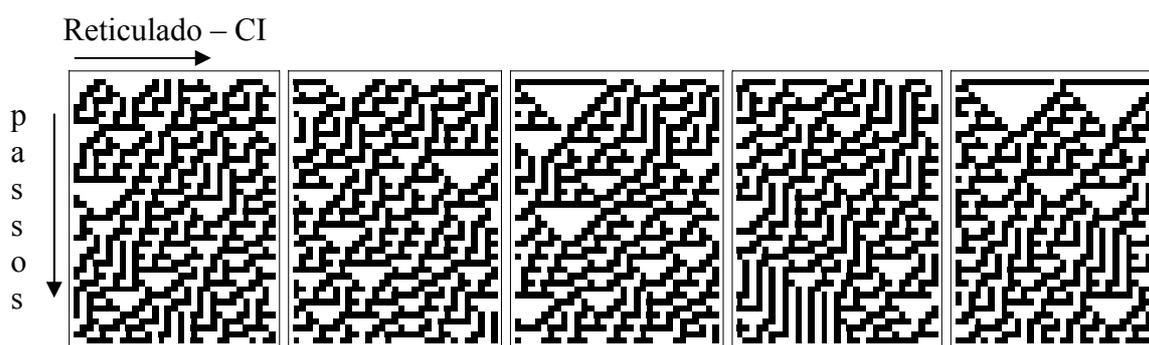


Figura 2.7: Representações da regra 30 sobre condições iniciais binárias aleatórias.

**Classe 4:** A classe 4 está entre a classe 2 e 3, sendo caracterizada por regiões que estão aparentemente misturadas de forma aleatória em regiões estacionárias, com

estruturas localizadas, ou que se movimentam linearmente em um sentido único, formando figuras com traços seqüenciais. Algumas configurações iniciais resultam em estruturas localizadas complexas, algumas vezes bastante duradouras. A regra 110 é um representante típico desta classe, como ilustra a Figura 2.8.

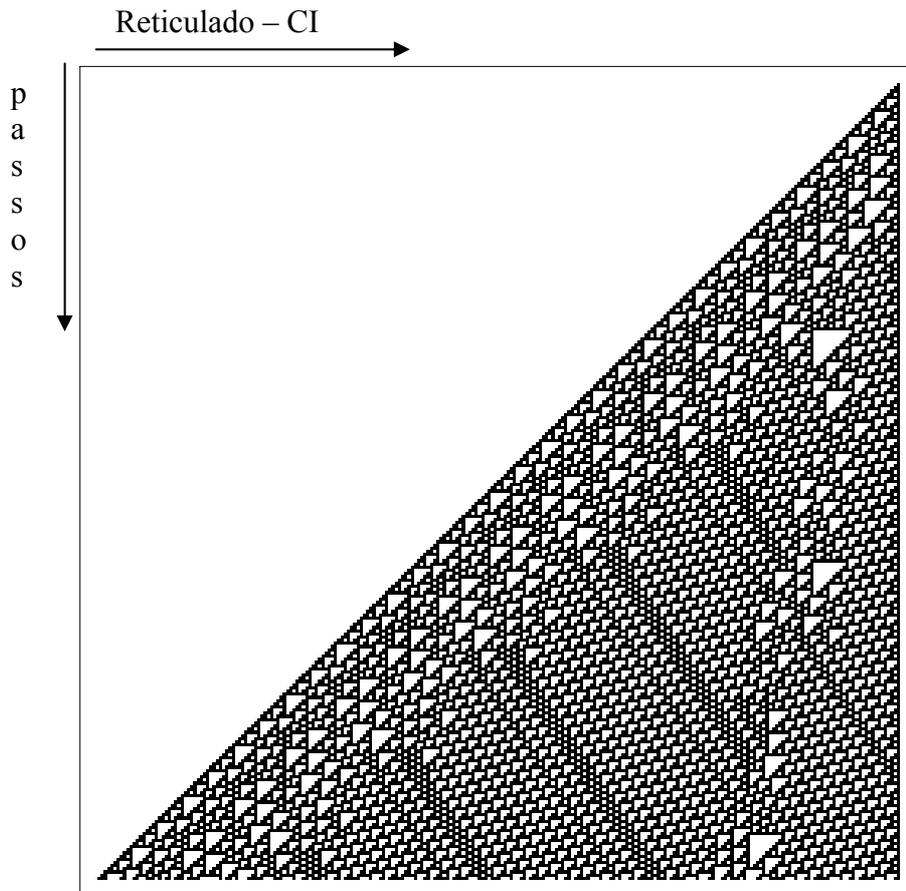


Figura 2.8: Representação da regra 110 aplicada durante 250 passos.

### 2.2.2. Computação Implícita e Explícita em ACs

Desde o princípio o desenvolvimento de autômatos celulares busca as relações entre o comportamento dinâmico genérico de um AC e suas possibilidades computacionais, como parte de uma questão maior sobre as relações entre a teoria dos sistemas dinâmicos e a teoria computacional [Wolfram, 2002]. Nessa perspectiva, pode-se dividir a forma de computação estudada nos ACs em duas forma: *computação explícita* e *computação implícita* [Oliveira, 2003].

A *computação implícita* é aquela executada diretamente da regra do AC, onde a configuração inicial do reticulado é o parâmetro de entrada para o processamento, e a

configuração final da aplicação da regra sobre o reticulado, após um número discreto de passos, é a saída do sistema.

A *computação explícita* é gerada pelo comportamento geral do AC onde, o papel da regra do AC é prover a dinâmica de sustentação sobre a qual a computação será executada. Neste caso, a entrada da computação não é toda a configuração inicial do reticulado, mas apenas uma parte dela, buscando-se encontrar modelos que associem a dinâmica de ACs com a máquina de Turing universal. Um exemplo de aplicação de computação explícita é o conhecido “Jogo da Vida”, criado por John Conway, que é capaz de simular uma máquina universal de Turing, gerando novas estruturas a partir de duas ou mais estruturas distintas, outro exemplo é a construção bidimensional de Alvy Smith que simula uma Máquina de Turing em tempo real [Mitchell, 1996].

Certas tarefas que teriam solução trivial na computação convencional ou na computação explícita a ACs, tornam-se extremamente complexas no contexto da computação implícita a regras de ACs, já que as mesmas devem ter como uma decorrência natural do seu comportamento dinâmico complexo, o poder computacional. Desta forma, as pesquisas neste tipo de abordagem buscam descobrir como os ACs podem ser utilizados como computadores paralelos rápidos e práticos [Toffoli, 1987], sem se preocupar em fazer uma analogia com a Máquina de Turing.

### 2.3. O problema de classificação da densidade

Por estar relacionado a vários trabalhos na busca de soluções através de computação implícita a regras de ACs, é útil considerar este problema em detalhe.

O problema de classificação da densidade (DCT), consiste em identificar qual o bit predominante em uma cadeia binária de comprimento ímpar.

Assim como para o problema da paridade, na DCT buscam-se ACs capazes de solucioná-lo através de computação implícita. Um AC unidimensional binário obtém sucesso na DCT quando:

- para um reticulado de tamanho ímpar  $e$ , quantidade de 1s superior a de 0s no conjunto inicial, este convergir todas as células do conjunto final para 1; ou,
- quando a quantidade de 1s do conjunto inicial for superior a de 0s, este convergir todas as células do conjunto final para 0.

Na Figura 2.9 é apresentado um exemplo da aplicação de uma regra de raio 3 sobre um reticulado de 37 células, durante 40 passos; enquanto no exemplo (a) tem-se uma quantidade de 1s superior a de 0s no conjunto inicial, no exemplo (b) tem-se quantidade de 0s superior a de 1s.

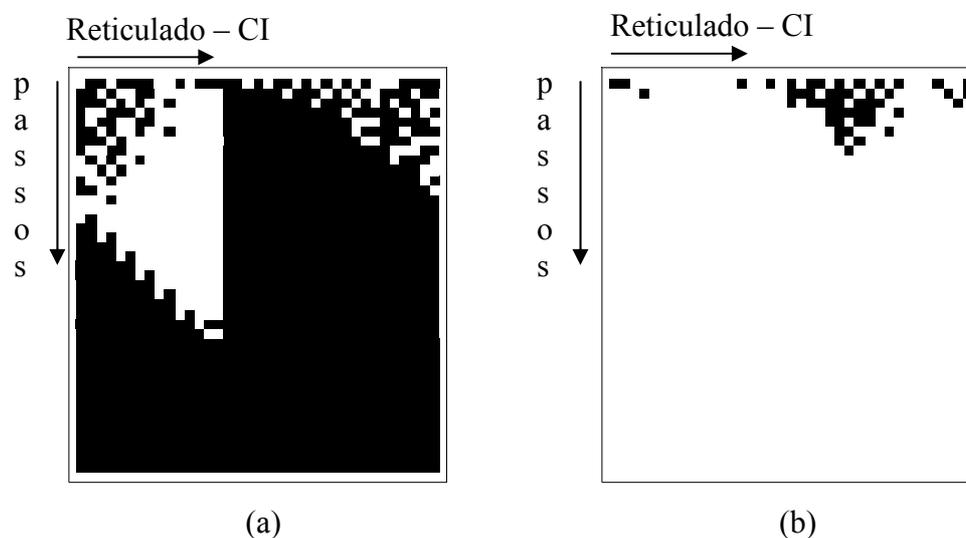


Figura 2.9: (a) Condição inicial com quantidade de 1s superior a de 0s; (b) . Condição inicial com quantidade de 0s superior a de 1s

## 2.4. O problema da paridade

Como no problema da classificação de densidade estudado em [de Oliveira, Bortot e Oliveira, 2006], [Juillé e Pollack, 1998], [Fuk s, 1997] e [Mitchell, 1993] que demonstra um seq encia de progressos na busca por regras de sucesso comprovado na solu o da DCT; o mesmo ocorre com o problema da paridade que, desde [de Oliveira e Vaiano, 2005], e do in cio deste trabalho, sofreu uma evolu o razo vel, ap s a aplica o de t cnicas estudadas em [Wolz e de Oliveira, 2007] que reconhece a maior dificuldade em se obter boas regras para o PP, se comparado com a DCT. A tabela a seguir mostra algumas das regras de excelente n vel de sucesso, para reticulados de comprimento  mpar, relatadas em [Wolz e de Oliveira, 2007], estas regras foram testadas em dois processos seq enciais e independentes com popula es de 400 bilh es de CIs geradas aleatoriamente, e com reticulado de comprimento  mpar variando de 37   149 c lulas.

<b>N�mero da regra avaliada</b>	<b>N�vel de sucesso</b>
328631059793571276683466076712148601604	100%
297244200729091206820116207266435155876	100%
281712218893097241562999445877824932304	100%

Tabela 2.1: Regras de excelente n vel de desempenho, para reticulados de comprimento  mpar relatadas em [Wolz e de Oliveira, 2007].

Esta dificuldade em se obter boas solu es para o PP est  relacionada ao fato de que o comportamento da regra do AC deve ser extremamente sens vel   varia o de um  nico bit na condi o inicial, j  que essa altera o deve refletir no comportamento limite esperado do AC. Mencione-se, entretanto, que em [Lee, Xu e Chau, 2001] provou-se a possibilidade de solucionar o PP atrav s da aplica o de uma s rie de regras elementares de ACs, com caracter sticas espec ficas, numa seq encia pr -definida. A Figura 2.10

apresenta o diagrama de um reticulado de 12 células, com um total de 3 células com valor 1 na condição inicial (CI), indicando paridade ímpar. Após um certo número de passos, e a aplicação de uma seqüência pré-definidas de regras simples de raio 1, todas as células convergem para 1.

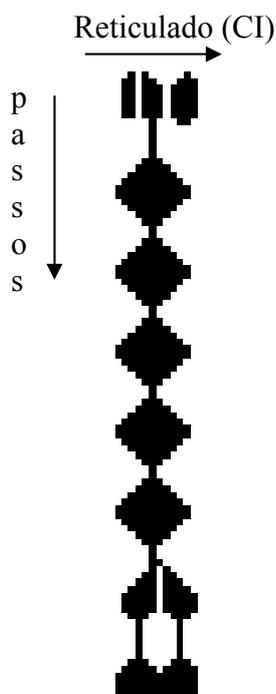


Figura 2.10: Demonstração de solução do problema da paridade conforme [Lee, Xu e Chau, 2001].

Em [de Oliveira e Vaiano, 2005] conduziu-se pesquisa que deu origem à presente, usando-se computação evolutiva para buscar autômatos celulares binários unidimensionais de raio 3 que solucionassem o problema da paridade. Nela foram utilizadas variações no processo de seleção do algoritmo evolutivo, na distribuição das configurações iniciais empregadas para avaliar as regras candidatas, e no método de avaliação em si. No entanto, nenhuma das tentativas foi efetiva no sentido de levar o processo evolutivo a uma regra satisfatória para o PP; de fato, como conclusão daquele

trabalho, sugere-se mesmo que, possivelmente, não exista uma regra de AC capaz de solucionar o problema.

Esta conclusão deve ser revista com base em resultados muito recentes [Wolz e de Oliveira, 2007], onde são representadas regras de desempenho praticamente perfeito na solução do PP, para reticulados de comprimento ímpar, e de sucesso alto (da ordem de 85%) para reticulados de comprimento par de tamanhos reduzidos (de 12 a 24 células), ainda que estas, porém, não apresentem robustez para comprimentos maiores. Estes resultados serviram como um indicativo da possível existência de uma, ou mais regras, dentro da população de regras elementares de raio 3, capazes de solucionar o problema da paridade; porém, a forma de obtenção destas regras e a interessante característica de solução, somente sobre reticulados ímpares, nortearam e serviram como item motivador para os passos seguintes deste trabalho.

A solução do PP, através das regras obtidas em [Wolz e de Oliveira, 2007], diferencia-se pelo fato de demonstrarem que existem regras especialistas em determinados comprimentos de reticulado; nesse sentido, destacam-se as que solucionam o problema para reticulados de tamanho ímpar, como demonstrado no exemplo da Figura 2.11, onde a regra 280363545044958142736483883671228394256 é capaz de solucionar o PP sobre um reticulado de comprimento ímpar (37 células) para 100 passos de execução do AC. No entanto, ela não obtém o mesmo sucesso sobre um reticulado de comprimento par (38 células). Observe que para a condição inicial (CI) de paridade par, num reticulado de comprimento par, todas as células convergem para 1s, invertendo o resultado esperado de acordo com a definição do problema, demonstrando o alto nível de especialização da regra em questão.

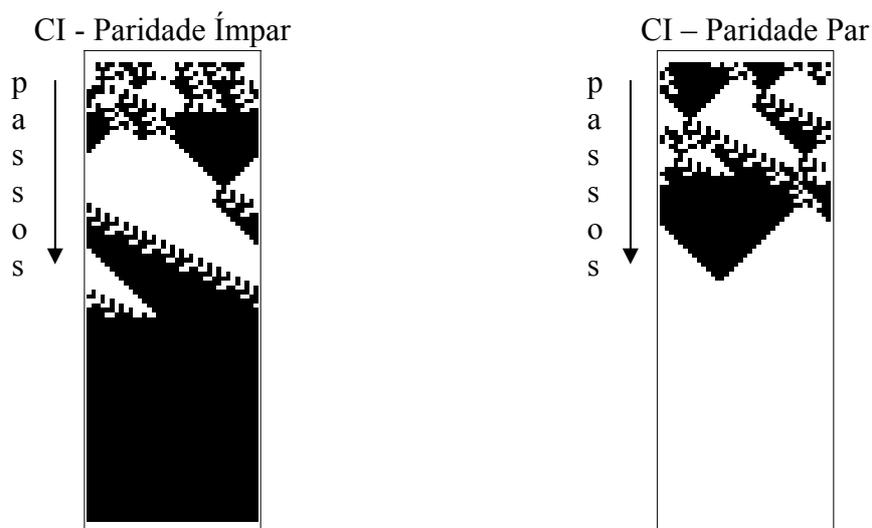


Figura 2.11: Aplicação da regra 280363545044958142736483883671228394256 sobre um reticulado de comprimento ímpar.

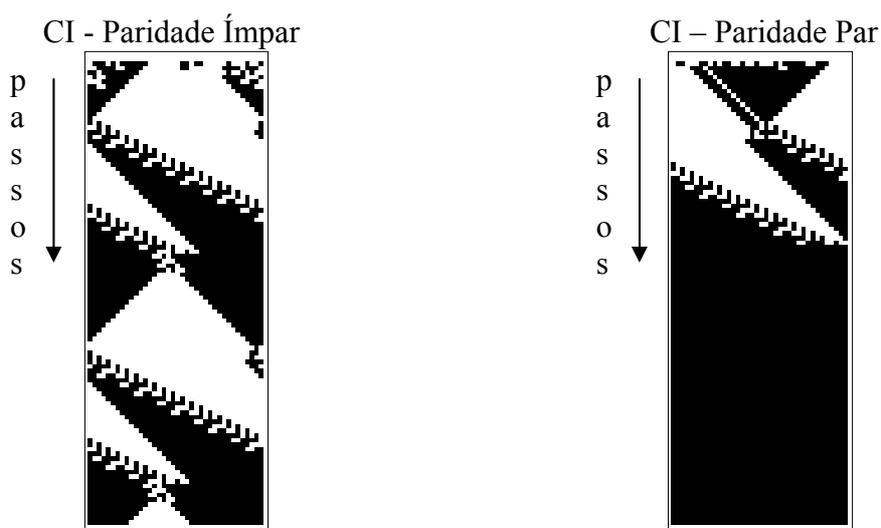


Figura 2.12: Aplicação da regra 280363545044958142736483883671228394256 sobre um reticulado de comprimento par.

Como teste de validação, esta regra foi submetida a uma série de 1000 CIs de reticulado com comprimento ímpar de 55 células, geradas de forma aleatória, e aplicada durante um período mínimo de 200 passos de execução (iterações) sobre cada configuração inicial, e o mesmo foi realizado para 1000 conjuntos iniciais com um

reticulado de comprimento par de 54 células. As condições iniciais foram obtidas de forma aleatória através de distribuição binomial, sendo que o mesmo conjunto de testes também foi realizado para condições iniciais obtidas através de distribuição uniforme. Em todos os testes foi obtido um nível desempenho próximo de 100% de sucesso, em reticulados de comprimento ímpar.

## 2.5. Computação evolutiva

O termo computação evolutiva refere-se aos processos computacionais que buscam imitar, de alguma forma, os mecanismos de evolução biológica.

Nos processos de computação evolutiva são utilizados alguns pontos básicos como:

- Uma população inicial de soluções candidatas com as características necessárias ao processo evolutivo, normalmente geradas de forma aleatória.
- Uma função de avaliação, capaz de julgar a aptidão de cada um dos indivíduos.
- Geração de mecanismos, aplicados à população de uma geração, que possibilitam obter os indivíduos da próxima geração.

Dentre os vários processos evolutivos conhecidos destaca-se, neste trabalho, a utilização de algoritmos genéticos [Holland, 1975], cujo mecanismo será melhor descrito no capítulo a seguir.

Uma evolução natural implica que os organismos devam se adaptar a seu meio ambiente, o que ocorre através de um trabalho evolucionário que percorre várias gerações. O termo algoritmo genético (AG) [Koza, 1993] descreve basicamente uma técnica computacional inspirada nos princípios genéticos que buscam simular os processos evolutivos em um nível genético [Jacob, 2001].

O objetivo deste tipo de abordagem é gerar uma implementação robusta de sistemas adaptativos, seguindo os paradigmas conhecidos, utilizados pela natureza, na evolução da estrutura genética dos seres vivos.

Tais algoritmos são inspirados em princípios evolutivos, em que indivíduos (representando soluções candidatas de um espaço de busca) são gerados através da alteração e recombinação da carga genética de dois ou mais indivíduos 'pais', e progressivamente passam a representar soluções candidatas cada vez melhores.

Em [Hillis, 1992] utilizou-se o conceito mais comum entre muitas espécies que se reproduzem sexuadamente, em que os genes de um dos conjuntos de cromossomas de uma célula diploide são herdados do gameta do pai, enquanto que os genes do outro conjunto são do gameta da mãe, porém, na maioria dos AG utiliza-se a representação de cromossomas aploides dentro de uma reprodução sexuada, padrão utilizado no processo evolutivo descritos neste trabalho.

No presente trabalho o processo de busca é iniciado com um conjunto de representações de regras de ACs geradas aleatoriamente e, ao longo da evolução, as regras vão sendo criadas. A cada geração regras são selecionadas e avaliadas de acordo com uma função que permite verificar sua aptidão (*fitness*) em solucionar o problema proposto no caso, o problema da paridade.

A seleção das regras segue um padrão de elitismo que consiste em preservar uma maior percentagem da população de melhor aptidão permitindo que a carga genética de 'melhor qualidade' desses indivíduos permaneça no processo evolutivo. O percentual restante dos indivíduos que não fizeram parte desta elite pode ser gerado de forma aleatória, ou seguindo um padrão de geração pré-definido.

Parte desta população ainda poderá sofrer um certo grau de mutação, que consiste em alterar, ou inverter, parte da carga genética dos indivíduos. A probabilidade de mutação pode ser quantificada da seguinte forma: para a probabilidade  $p_m$  de não ocorrer nenhuma mutação dos bits do cromossomo  $H$ , e  $S_m(H)$  como sendo a probabilidade do cromossomo  $H$  sobreviver a uma mutação tem-se que  $S_m(H) = (1 - p_m)^{O(H)}$ , para  $O(H)$  sendo o número de bits definidos em  $H$  [Mitchell, 1998].

Dessa forma, espera-se que as regras de maior sucesso sejam derivadas de um desenvolvimento genético entre várias gerações que, naturalmente, buscam a solução e adaptação ao 'ambiente' que lhes é apresentado [Mitchell, 1996].

## Capítulo III: Experimentos realizados

Inicialmente, foram conduzidos experimentos buscando dar continuidade aos estudos realizados em [de Oliveira e Vaiano, 2005]. Neste intuito foram geradas variações nas distribuições que formam as condições iniciais dos autômatos, utilizando-se de distribuições pouco exploradas em [de Oliveira e Vaiano, 2005], e de generalizações das distribuições binomial e uniforme.

Após o sucesso das regras obtidas segundo a metodologia discutida em [Wolz e de Oliveira, 2007] passou-se a testar algumas das variações lá aplicadas no processo evolutivo. Inicialmente buscou-se verificar o efeito da variação do comprimento do reticulado, e como a aplicação de reticulados com comprimentos pares e ímpares, afetam o processo evolutivo. Em seguida, verificou-se a eficácia de uma nova metodologia de cruzamento genético, derivada de [Franks, 2004].

Na seqüência buscou-se reproduzir alguns aspectos dos experimentos realizados em [Wolz e de Oliveira, 2007] permitindo uma análise individual das variáveis que influenciaram nos resultados lá obtidos; primeiramente foram ampliados os parâmetros da pesquisa, com maiores valores do tamanho da população e do grau de elitismo, com CIs de comprimento ímpar, variando seu comprimento através de intervalos variáveis ao longo das gerações. Por fim, foi testada a introdução da chamada “pré-avaliação” das CIs, proposta em [Wolz e de Oliveira, 2007].

### 3.1. Parâmetros base

Os experimentos realizados nas Seções 3.2 a 3.6 utilizam alguns parâmetros base para sua execução, que são descritos nesta seção.

Nos experimentos conduzidos são utilizadas regras de ACs unidimensionais binários de raio 3 (que representa um conjunto de  $2^{128}$  regras candidatas), e a avaliação de cada uma se dá com base em 250 configurações iniciais (CIs) a ela submetidas, geradas aleatoriamente a cada geração, segundo determinada distribuição. O processo evolutivo transcorre então por no mínimo 60 gerações, com população de 100 regras, sendo que a inicial é gerada aleatoriamente sobre todo o espaço de busca. Outros aspectos dos experimentos são os seguintes:

- *Função de avaliação:* A aptidão de cada regra é medida sobre cada conjunto das 250 configurações iniciais a ela submetidas. Após um período mínimo de 150 passos de execução (iterações) sobre cada configuração inicial, a regra avaliada recebe pontuação de 1 unidade se ela leva todas as células do reticulado para 0 ou para 1, de acordo com a paridade da configuração inicial; caso contrário, a avaliação da regra é nula para a configuração inicial em questão. A aptidão da regra em uma geração é dada então pela soma dos pontos obtidos individualmente para cada configuração inicial.
- *Método de Seleção:*
  - *Elitismo:* Buscando manter certa carga genética dos melhores indivíduos avaliados, utilizou-se elitismo, preservando-se os 20 melhores indivíduos de uma geração para a seguinte, sem que estes sofram cruzamento ou mutação.
  - *Seleção aleatória a partir da elite:* Duas formas foram empregadas para o preenchimento do restante da população, ambas tendo como base a elite: seleção puramente aleatória e seleção por roleta.

- *Cruzamento*: As soluções candidatas restantes são geradas através do cruzamento da carga genética e da mutação das melhores soluções selecionadas anteriormente.
- *Mutação*: Após o cruzamento, todas as regras ficam sujeitas a uma possibilidade aleatória de mutação de 2% por bit, por indivíduo.
- *Condição inicial com distribuição binomial*: Nesta distribuição de CIs a quantidade de 1s apresenta distribuição binomial, o que, naturalmente, ocasiona probabilidade maior de que as CIs possuam quantidade de 1s próxima da quantidade de 0s, próximo, portanto, do chamado *balanceamento* da CI.
- *Condição inicial com distribuição uniforme*: Nesta distribuição, todas as quantidades de 1s e 0s são passíveis e equiprováveis, nas CIs.

Inicialmente, e apenas para aferição, foram executados os mesmos experimentos de [de Oliveira e Vaiano, 2005], variando-se o método de seleção, entre a seleção puramente aleatória a partir da elite, e a por roleta sobre a elite. Os resultados obtidos (utilizando CIs com distribuição binomial e uniforme) são exemplificados no gráfico da Figura 3.1, onde se tem no eixo X o número de gerações, e no eixo Y a aptidão atingida pelas regras a cada geração, mostrando o desempenho das melhores e piores regras, e as médias de desempenho das regras a cada geração. Observa-se que o resultados das melhores regras é próximo a 140 com média próxima a 120, o que representa nível máximo de sucesso de 56%. Em outras palavras, a melhor regra obteve sucesso no reconhecimento de 140 condições iniciais das 250 propostas. Este resultado, e dos outros experimentos, foram constatados após um mínimo de 20 execuções, 10 para cada método de seleção.

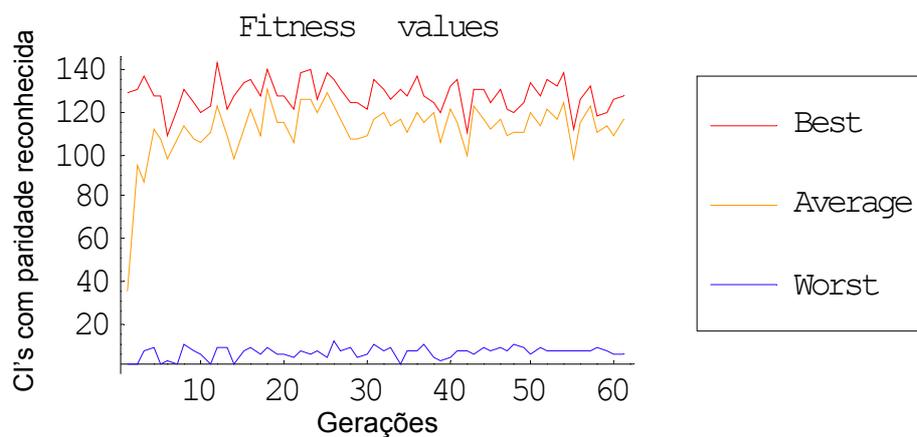


Figura 3.1: Exemplo de gráfico de CI's com distribuição uniforme.

As mesmas variações de métodos de seleção foram aplicadas utilizando-se configurações iniciais com distribuição binomial, sendo atingido o mesmo nível máximo de sucesso (próximo de 56%, i.e., 140 regras reconhecidas).

### 3.2. Variação do número de gerações

A princípio buscou-se aumentar o número de gerações, de 60 para 240, utilizando-se as mesmas distribuições iniciais (uniforme e binomial) para geração das CI's, na intenção de avaliar se em [de Oliveira e Vaiano, 2005] o processo evolutivo não teria sido interrompido muito cedo, antes de obter uma carga genética adequada para o PP.

Constatou-se que os valores obtidos permaneceram nos mesmos patamares de outras distribuições já avaliadas em [de Oliveira e Vaiano, 2005].

### 3.3. Distribuição quase-balanceada

Denominou-se *quase-balanceada* a distribuição em que todas as configurações iniciais amostradas tenham proporção de 1s e 0s variando em somente uma unidade.

Neste experimento foram utilizadas CIs de distribuições quase-balanceada com 90% de probabilidade de um 1s, em um reticulado de 35 células.

Os resultados obtidos, diferentemente da distribuição binomial, não demonstraram variações no processo evolutivo nem positivas, nem negativas. Observou-se que o processo evolutivo ainda tende à obtenção de regras de pouca robustez, e com desempenho que possuem um desempenho superior somente para condições extremas, onde o reticulado é composto por uma quantidade elevada de 1s e 0s, nas outras condições o nível de sucesso não foi superior a 56% (obteve sucesso no reconhecimento de 140 CIs) [Silva e de Oliveira, 2006].

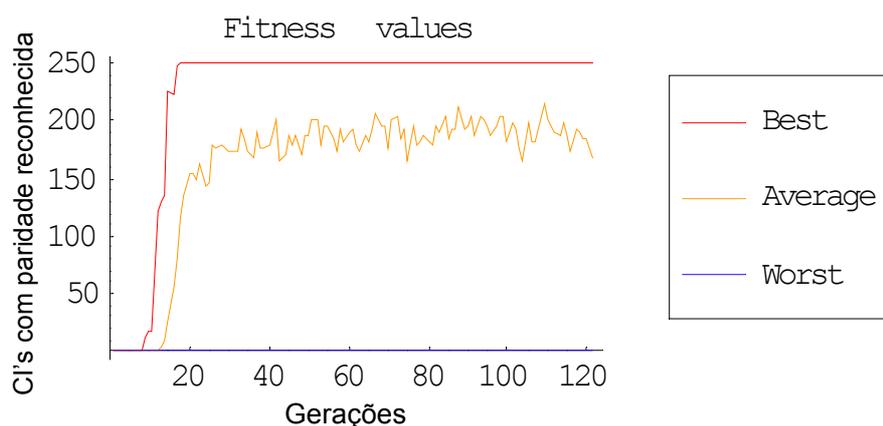


Figura 3.2: Distribuição quase balanceada com uma proporção de 90% de 1s.

### 3.4. Distribuição quase-balanceada com avaliação parcial

O esquema de avaliação parcial das regras consiste na definição de uma escala de pontuação de 0 a 1 que varia conforme o tamanho do reticulado. Sendo  $N$  o tamanho do reticulado, e  $S$  o número de subdivisões da escala de pontuação, tem-se  $N/S$  níveis de avaliação possíveis, com  $S = N$ , basta que um único bit correto apareça na configuração final para que a regra deixe de receber avaliação nula; já com  $S = 1$ , tem-se a situação original (não parcial) de avaliação.

Buscou-se verificar o efeito de uma avaliação parcial das regras que, diferentemente do método convencional definido na função de avaliação, a regra recebe uma pontuação intermediária, mesmo não sendo capaz de levar 100% das células do reticulado para 0 ou 1. Desta forma a pressão seletiva do processo evolutivo fica reduzida, havendo maior chance de aproveitamento da carga genética das regras relativamente menos eficazes.

Foram executados testes para S valendo 1, 2 e 11. Em todos eles foram obtidos os mesmos resultados. A aplicação desta metodologia provocou uma lentidão no processo evolutivo, e terminando no mesmo nível de sucesso atingido com configurações iniciais quase-balanceadas sem a aplicação de avaliação parcial.

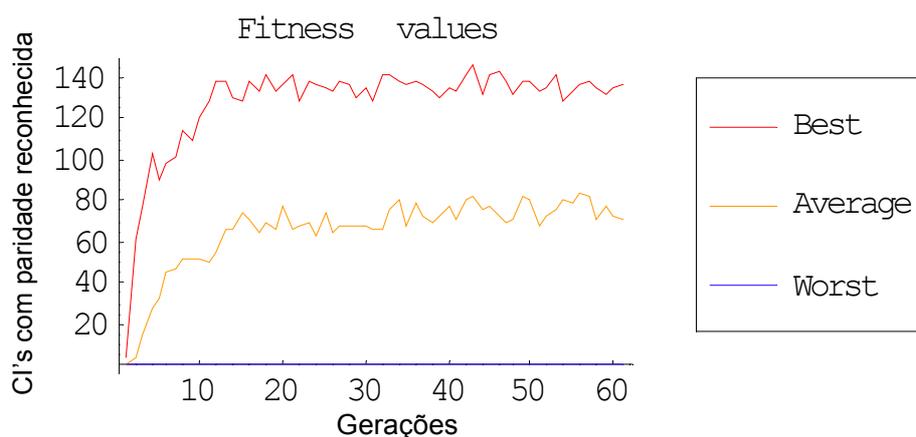


Figura 3.3: Exemplo de gráfico de CIs com distribuição quase-balanceada e avaliação parcial.

### 3.5. Configurações iniciais com distribuições generalizadas

Novos tipos de distribuição das configurações iniciais foram criadas, generalizando-se as distribuições binomial e quase-balanceada. Assim, no caso da distribuição binomial generalizada, em vez de se ter a máxima probabilidade de 1s das CIs em 50%, pode-se deslocar essa proporção arbitrariamente para mais ou para menos;

analogamente, para a distribuição quase-balanceada generalizada a diferença entre as quantidades de bits em uma CI pode ser alterada para outros valores exatos, adicionalmente à quantidade padrão de 1, que caracteriza a distribuição quase-balanceada mencionada anteriormente.

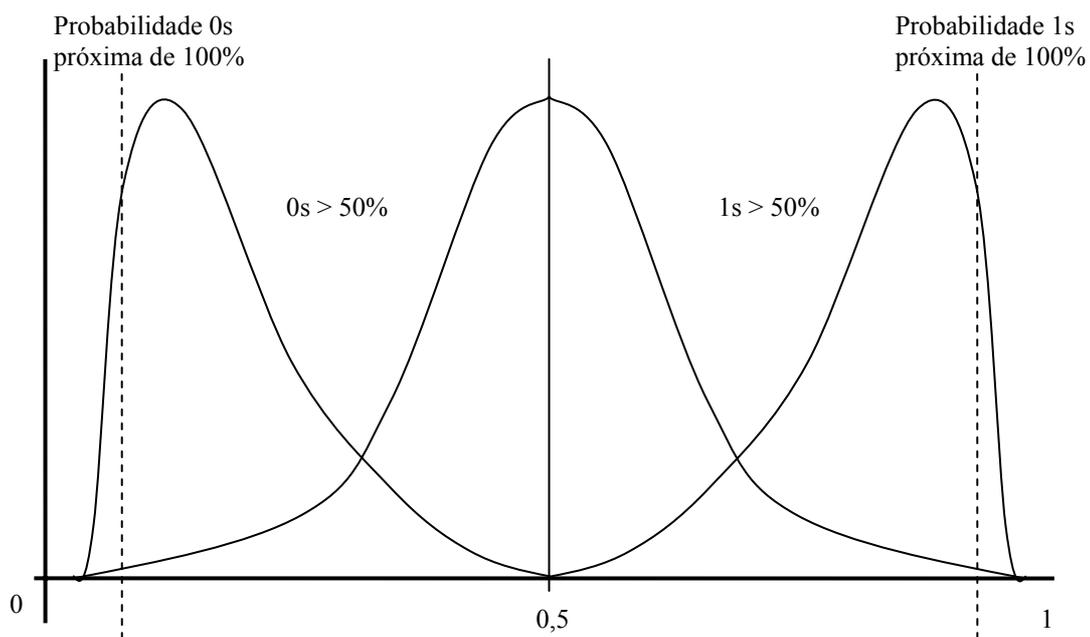


Figura 3.4: Gráfico de distribuição binomial com curvas generalizadas à direita e à esquerda.

Para ambos os tipos de distribuição generalizada verificou-se que, para densidades muito próximas de 1 ou de 0, o nível de sucesso das regras melhorava consideravelmente, como demonstra o gráfico da Figura 3.4.

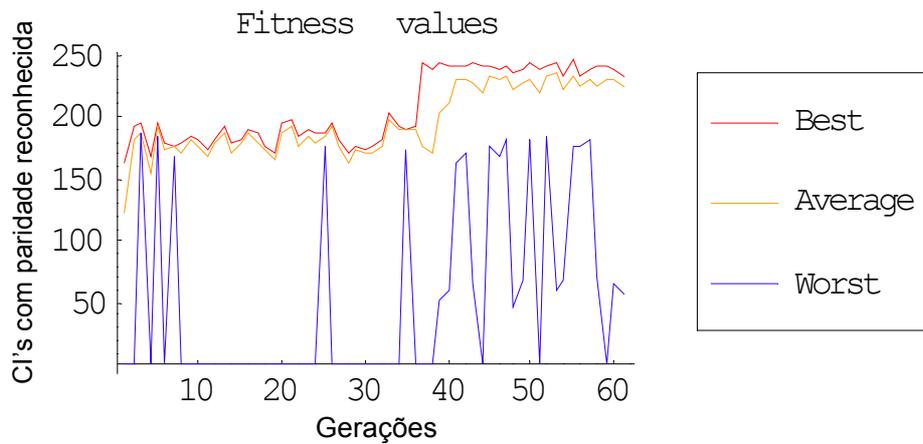


Figura 3.5: Aplicação de distribuições generalizadas.

No entanto, constatou-se que as boas regras nessa situação não têm interesse prático, dado que são pouco robustas, só resolvendo corretamente o problema da paridade para situações limites de densidade de 1s ou 0s, convertendo todas as células sempre para 1 ou para 0. Isso é ilustrado na Figura 3.5, com uma condição inicial de 37 células, com quantidade de 1s próxima de 100%; a evolução temporal mostrada ocorre por 60 iterações, e a regra é a de número 340274888262241388876171559378197828192.

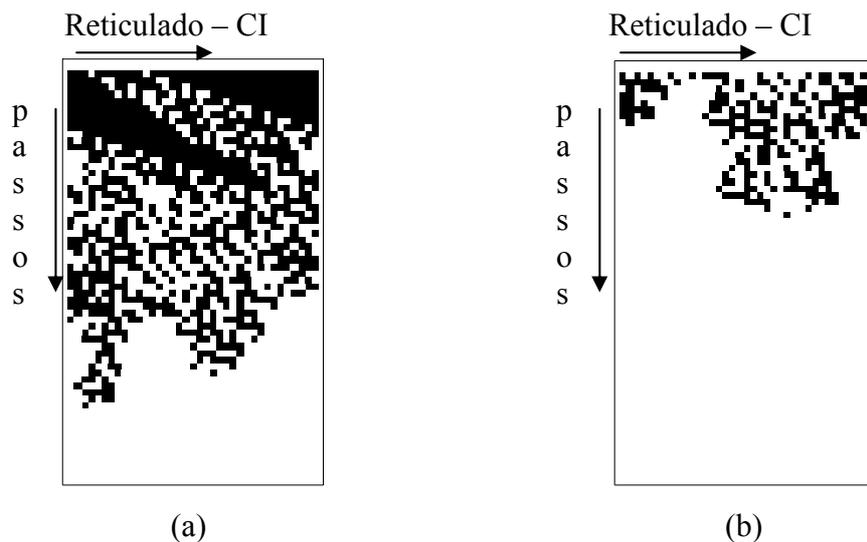


Figura 3.6: (a) CI com probabilidade de 1s próxima de 100% e paridade ímpar.  
(b) CI com probabilidade de 1s próxima de 50% e paridade par.

### 3.6. Reticulado variável

Visto que a variação das distribuições dos conjuntos iniciais pouco afetavam o processo evolutivo, focou-se nos estudos de [Wolz e de Oliveira, 2007] que estava obtendo boas regras com a implantação de várias alterações nos parâmetros do processo evolutivo. Uma destas alterações correspondia à variação do comprimento do reticulado durante o processo evolutivo. Para verificar o efeito que esta variação causaria no processo evolutivo, foram geradas e testadas diferentes situações.

A idéia aqui foi variar a dificuldade que as regras enfrentariam ao longo do processo evolutivo, por exemplo, de uma situação mais fácil (com reticulados pequenos) à mais complicada (com reticulados maiores); idéia análoga foi utilizada em [Breukelaar and Bäck, 2005], onde não se variou o tamanho de reticulado, mas sim suas distribuições de probabilidade.

Geraram-se então CIs com reticulados de tamanho variável em intervalos pré-definidos, isto é, o conjunto de CIs de avaliação criados a cada geração, foi subdividido igualmente entre os tamanhos pré-definidos no intervalo, obtendo-se assim um conjunto de CIs de tamanhos variados, cada um deles sujeito a uma distribuição binomial.

#### 3.6.1. Reticulado variável após 30 gerações

No primeiro experimento realizado variando-se o tamanho do reticulado, manteve-se fixo o tamanho do reticulado em 15 células, durante 30 gerações, e, em seguida, ele se tornou variável no intervalo de 15 a 37 células. Um dos resultados obtidos, típico do elenco de execuções, é visto abaixo:

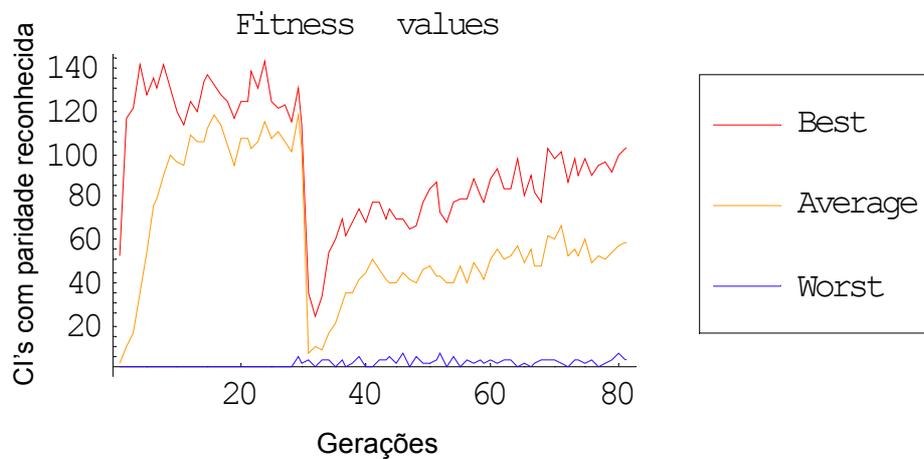


Figura 3.7: Evolução com reticulado variável após a 30ª geração.

Observa-se uma rápida evolução inicial, porém, no momento em que se varia o tamanho do reticulado há uma nítida queda no rendimento das regras; a partir de então a evolução prossegue, mas agora de forma mais compassada. Este processo evolutivo específico mostrado foi executado por mais 60 gerações, mas sem melhora. Em resumo, todas as execuções levaram aos valores já atingidos anteriormente, i.e., de cerca de 56% de sucesso.

### 3.6.2. Reticulado variável desde a primeira geração

Neste experimento o reticulado passa a ser variado (no mesmo intervalo do caso anterior, de 15 a 37 células) desde a primeira geração, produzindo uma evolução inicial mais lenta que, após uma média de 10 gerações, passa a se caracterizar como as outras execuções.

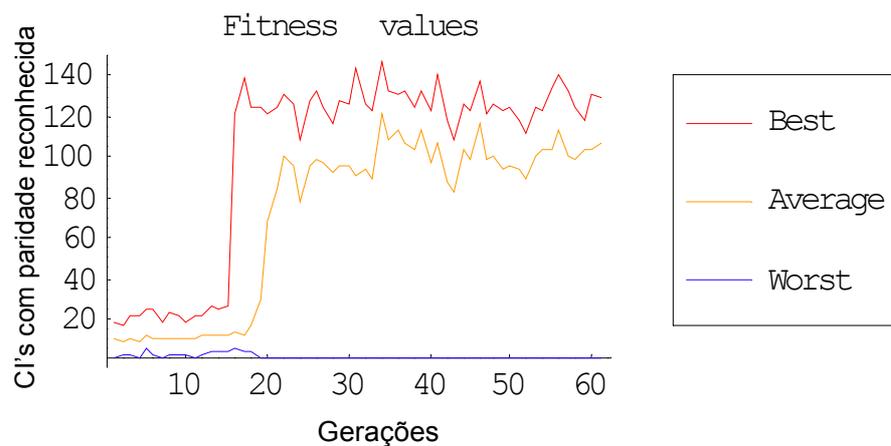


Figura 3.8: Evolução com reticulado variável a partir da primeira geração.

Este mesmo experimento foi executado por mais 300 gerações no intuito de checar se havia alguma variação, porém, não foi obtido o sucesso esperado, como mostra-se a seguir.

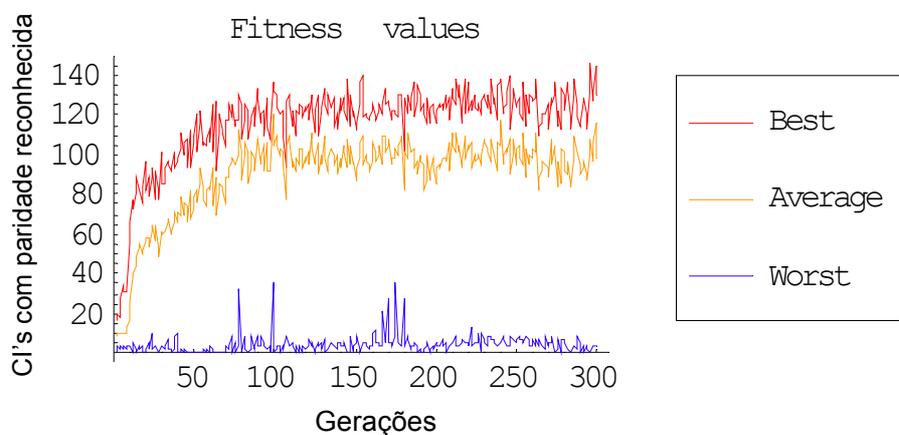


Figura 3.9: Evolução com reticulado variável a partir da primeira geração, durante 300 gerações.

### 3.6.3. Reticulados com comprimento ímpar

Aqui esperava-se obter as mesmas regras obtidas em [Wolz , de Oliveira, 2007] especializadas em reticulados ímpares, e com níveis de sucesso elevados. A idéia era

verificar se somente a variação do reticulado, com comprimentos ímpares, poderia ser suficiente para afetar o processo evolutivo e gerar boas regras.

Desta forma executou-se uma variação do comprimento do reticulado apenas com quantidades ímpares de células, no caso, todos e apenas os tamanhos de reticulado de comprimento ímpar, entre 15 e 37 células.

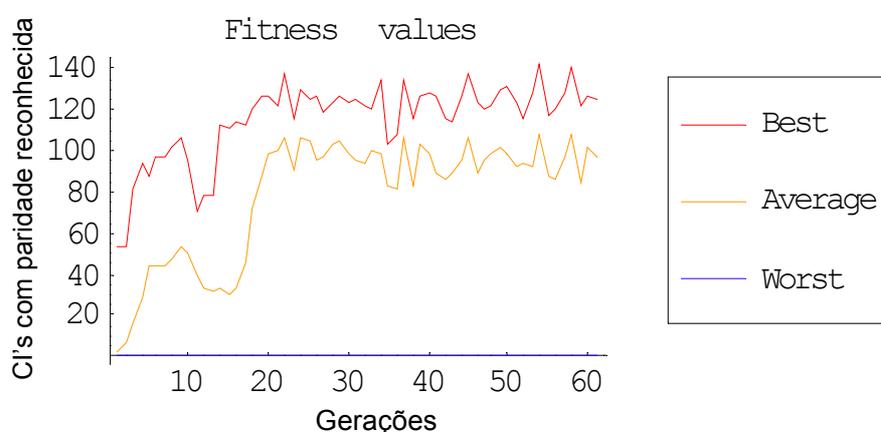


Figura 3.10: Evolução com reticulados de comprimento ímpar.

O resultado obtido não foi diferente do obtido com reticulado variável.

#### 3.6.4. Reticulados com comprimento par

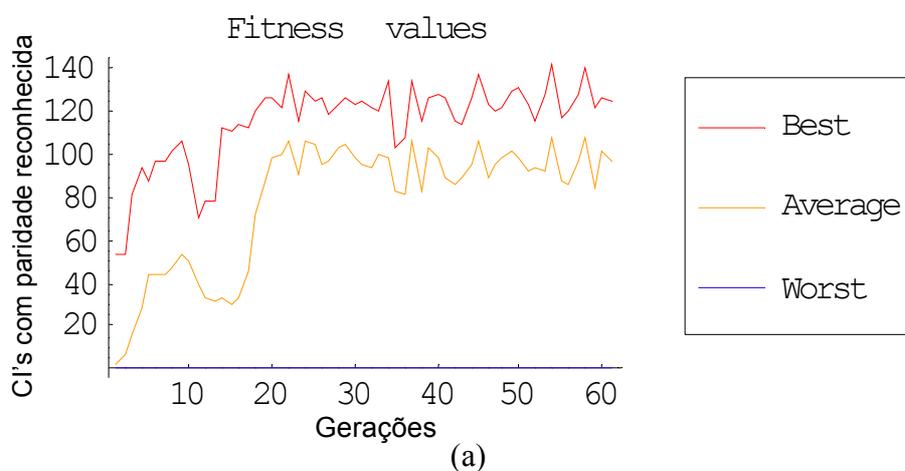
O mesmo experimento anterior foi realizado, mas agora apenas para os tamanhos de reticulado com comprimento par, no intervalo de 16 e 36 células.

E o resultado foi o mesmo do encontrado no reticulado de tamanho ímpar, evidenciando que a paridade do número de células do reticulado não afetou o processo evolutivo em questão.

### 3.6.5. Reticulado com comprimento crescente por intervalo de gerações

Nesta avaliação foram gerados reticulados com tamanhos variáveis a cada intervalo definido de gerações, sendo que o tamanho do reticulado variava conforme um passo crescente de células. Assim, objetivando-se que, ao final de 60 gerações, a variação do comprimento do reticulado estivesse ocorrendo no intervalo de 15 a 37 células, estabeleceu-se que da 1ª até a 3ª geração os reticulados teriam de 15 a 17 células e, a partir daí, o comprimento superior seria acrescido de 2 células a cada 3 gerações. Desta forma no intervalo entre a 3ª e a 6ª gerações, os comprimentos dos reticulados variariam entre 15 e 19 células, e assim por diante, até que no intervalo entre a 57ª e 60ª gerações, seria obtida a variação especificada de 15 a 37 células.

Esta metodologia foi executada com variações a cada 3, 10 e 20 gerações. O resultado de todos esses experimentos foi uma rápida evolução inicial, com queda de desempenho no momento em que o tamanho do reticulado era alterado, e voltando aos mesmos níveis de sucesso já obtidos até então, da ordem de 56%, sendo que, em alguns casos, observou-se uma (provavelmente temporária) regressão do processo evolutivo.



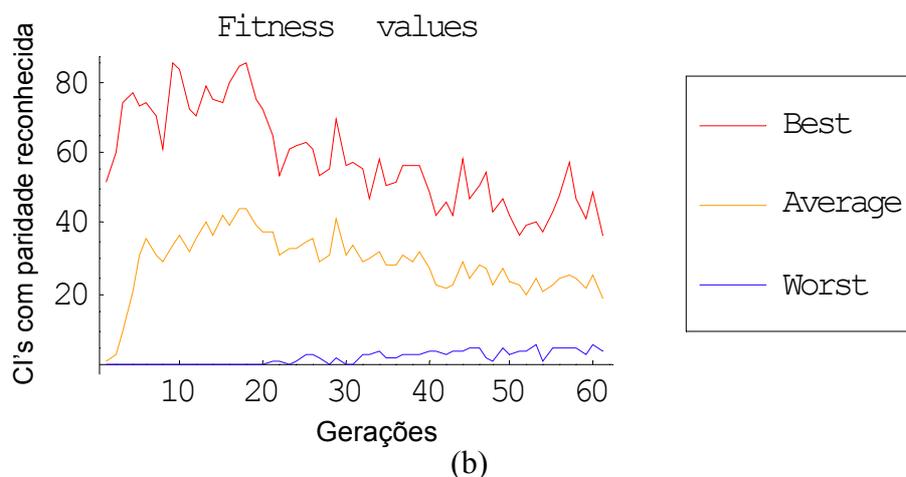


Figura 3.11: (a) Evolução para intervalos crescentes a cada 10 gerações;  
 (b) Regressão do processo evolutivo gerado por variação de intervalos crescentes a cada 10 gerações.

### 3.7. Um novo operador de recombinação

Visto que, somente a variação do reticulado não foi capaz de influenciar no desempenho do AG, foi estudada a possibilidade de introdução de ruído no sistema [Moreira, Mathur, Diemeier e Amaral, 2004], porém, como já havíamos utilizado algo similar na aplicação de uma avaliação parcial, optou-se por, testar a metodologia de recombinação utilizada em [Franks, 2004].

Neste esquema gera a recombinação através de um número inteiro  $j$ , gerado de forma aleatória e localizado entre 0 e  $k-1$ , sendo  $k$  o número de bits que formam a vizinhança; desta forma, permite-se o fracionamento do cromossomo, produzindo múltiplos pontos de cruzamento, a partir de combinações de blocos de tamanho  $2j$ , alternados, das duas regras a serem recombinadas.

Por exemplo, para  $k = 3$  e  $r = 1$ , tem-se  $j \in [0, 2]$ , o que gera blocos com 1, 2 ou 4 bits. Se o bit  $j$  da seqüência de saída é igual a 1, o filho usará a saída do cromossomo pai-

2, do contrário, utilizará a saída do cromossomo pai-1, concatenando de forma igualitária a carga genética.

**Cromossomo Pai - 1**

Vizinhança			Valor
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

↑  $j = 1$

**Cromossomo Pai - 2**

Vizinhança			Valor
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

↑  $j = 1$

Tabela 3.1: Representação da forma de fracionamento dos cromossomos para  $j = 1$ .

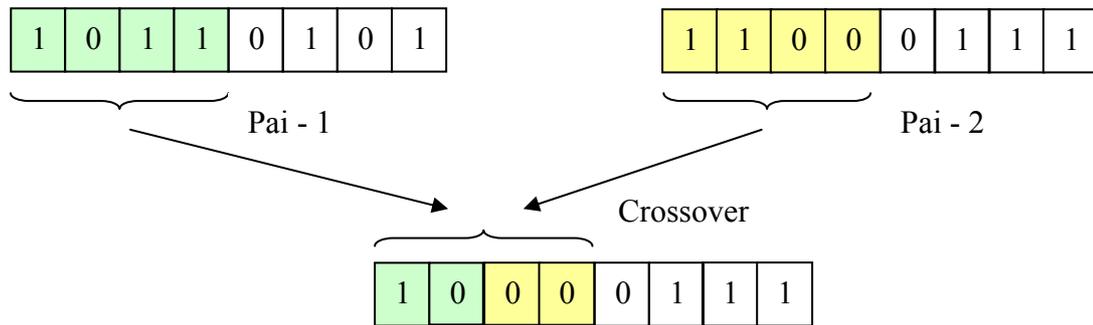


Figura 3.12: Representação do novo operador de recombinação.

Foram feitos testes para checar a validade desta idéia através de comparativos de resultados de um problema bem conhecido, a DCT, e os mesmos resultados foram obtidos com a recombinação de um ponto, todos consistentemente próximos de 80% de sucesso.

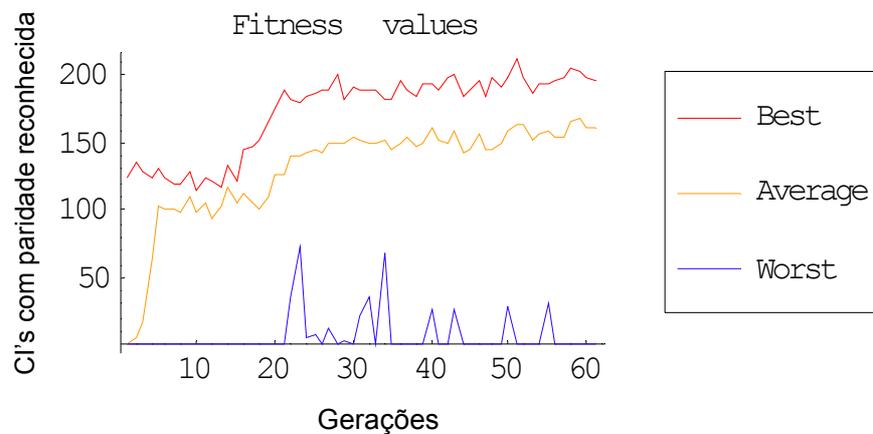


Figura 3.13: Teste do novo modelo de recombinação na DCT.

Como os testes conduzidos não trouxeram vantagens ao processo evolutivo para o caso da DCT, considerou-se que seu desempenho possivelmente também não seria efetivo ao PP. De qualquer forma, tal hipótese foi testada, conforme mostram os três experimentos a seguir, primeiro utilizando CIs com distribuição binomial, em seguida com CIs quase-balanceadas e, finalmente, com CIs de comprimento variado.

### 3.7.1. Novo operador de recombinação e CIs com distribuição binomial

Para o PP foram iniciados testes através da distribuição binomial. Os resultados obtidos foram inferiores apurados nos experimentos anteriores, já que das 10 evoluções realizadas apenas uma gerou um resultado próximo de 56% de sucesso, sendo que as outras levaram a resultados próximos de 48%.

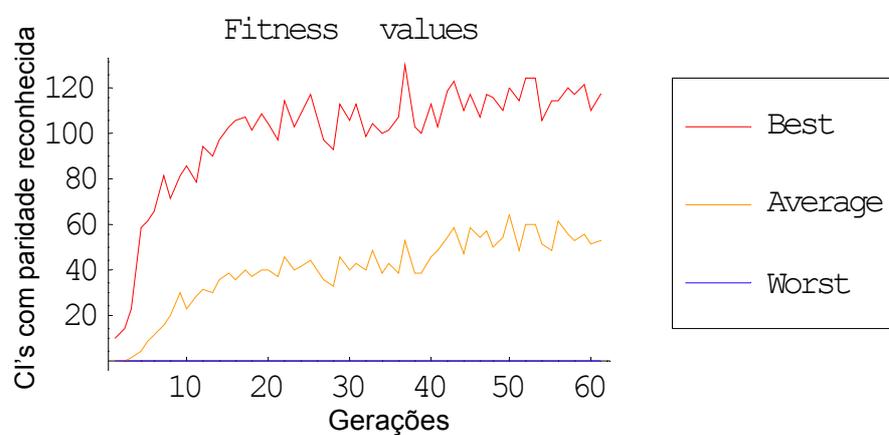


Figura 3.14: Novo operador de recombinação com distribuição binomial.

Percebe-se que houve uma dificuldade maior de evolução das regras. Para verificar se havia alguma tendência evolutiva após as 60 gerações, estendeu-se o processo a 300 gerações, porém, não foi encontrada nenhuma tendência evolutiva positiva.

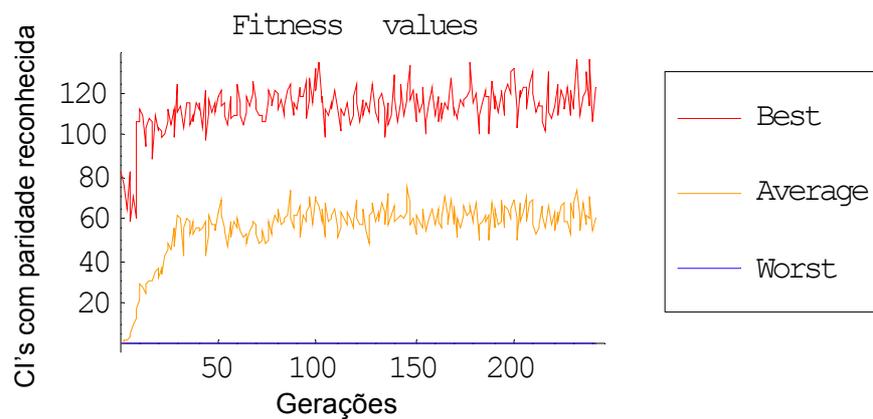


Figura 3.15: Novo operador de recombinação com distribuição binomial, por 300 gerações.

### 3.7.2. Novo operador de recombinação e CIs com distribuição quase-balanceada

Repetiu-se então experimento anterior, agora com CIs quase-balanceadas em um reticulado de 35 células. Os resultados obtidos, diferentemente do obtido com a distribuição binomial, não demonstraram variações no processo evolutivo, levando a evolução aos patamares típicos que vinham sendo obtidos no trabalho.



Figura 3.16: Novo operador de recombinação com distribuição quase-balanceada.

### 3.7.3. Novo operador de recombinação e CIs com comprimento crescente por intervalo de gerações

Devido aos fracos resultados obtidos com a distribuição binomial, optou-se por também testar o novo operador com essa distribuição, mas em reticulado de comprimento variável e crescente a cada 10 gerações, com a meta de que eles estivessem variando no intervalo de 15 a 37 células, ao final de 60 gerações.

Os resultados obtidos não se mostraram diferentes dos já obtidos com o operador de recombinação de um ponto.

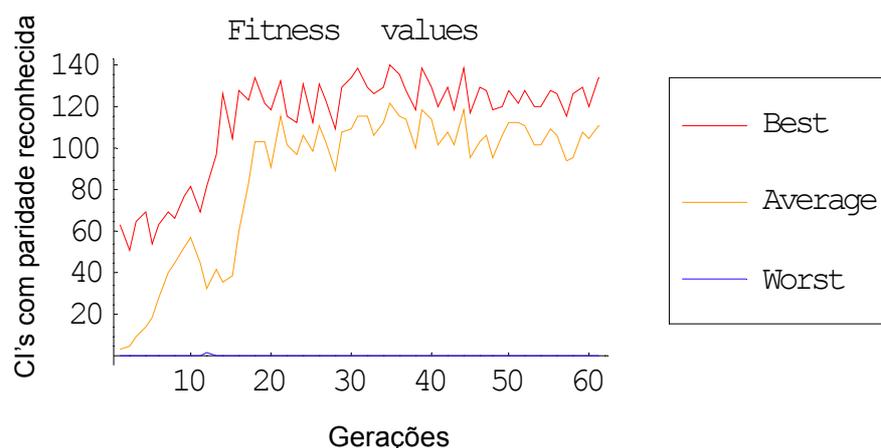


Figura 3.17: Novo operador de recombinação com reticulado com comprimento crescente por intervalo de gerações.

### 3.8. Novos parâmetros de busca

Os valores dos parâmetros utilizados nos experimentos anteriores eram baseados em [de Oliveira e Vaiano, 2005]. No entanto, face aos resultados infrutíferos obtidos, focou-se na tentativa de obtenção de regras de comprimento ímpar, conforme reportado em [Wolz, de Oliveira, 2007]. Assim, neste experimento, e nos seguintes, foram utilizados

novos valores em alguns dos parâmetros do processo evolutivo, segundo o estabelecido em [Wolz, de Oliveira, 2007], quais sejam:

- Grau de elitismo de 95%;
- População inicial de 500 indivíduos;
- Evolução por 500 gerações;
- Mutação de 4% por bit, por indivíduo;

As CIs foram geradas com intervalos crescentes de comprimento, e quantidades igualmente distribuídas para cada comprimento. As variações testadas são descritas a seguir.

### 3.8.1. Janelas progressivamente crescente

Foram criados grupos de CIs com 'janelas' de comprimentos progressivamente crescentes, iniciando com CIs de 15 a 17 células, e aumentando na extremidade maior, a cada 13 gerações. Ou seja:

- Inicia-se o processo com CIs de comprimento variando entre 15 e 19 células até a 13ª geração;
- Neste ponto o limite superior é acrescido de 3 unidades passando de 19 para 21, fazendo com que as CIs passassem a variar com comprimentos entre 15 a 21, e assim por diante;
- Esta seqüência de acréscimos do limite superior prossegue então até a última geração, quando os comprimentos variava de 15 a 51 células.

O resultado do experimento [Silva e de Oliveira, 2007] pode ser visto figura abaixo.

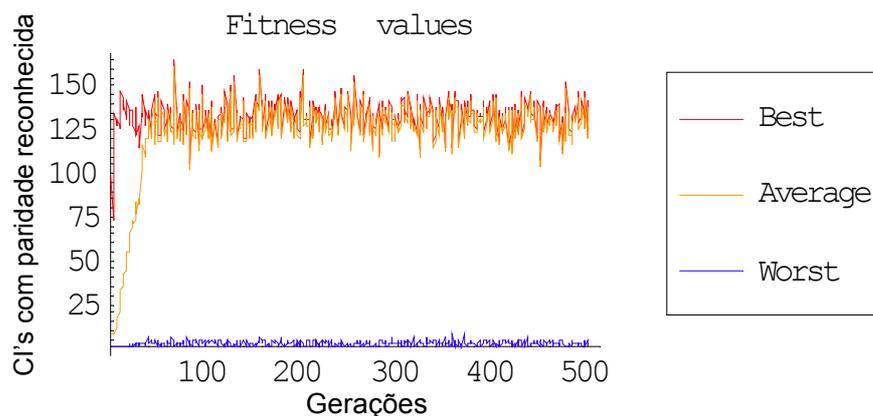


Figura 3.18: Janelas progressivamente crescentes

### 3.8.2. Janelas de tamanho praticamente fixos, com sobreposição de comprimentos

Este método foi o utilizado em [Wolz e de Oliveira, 2007] e baseia-se em uma série de grupos de comprimentos fixos durante um certo número de gerações, que variam de forma crescente, tanto nos valores limítrofes inferiores, quanto superiores. No experimento realizado iniciando o conjunto com comprimentos variando de 15 a 23 células, a cada 17 gerações ele foi alterado segundo a seqüência: de 17 a 25, de 19 a 27, etc, até comprimentos de 21 a 51 células.

Os resultados obtidos geraram evoluções como a apresentada a seguir.

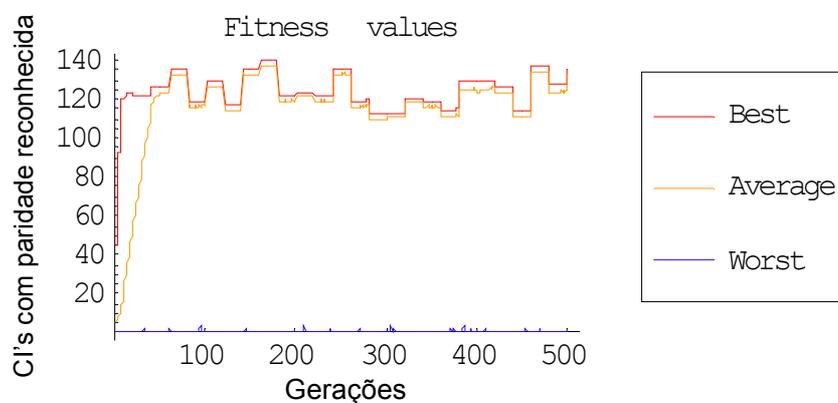


Figura 3.19: Janelas de tamanhos praticamente fixos, com sobreposição de comprimento.

Como pode ser observado nos gráficos das Figuras 3.18 e 3.19, não se obtiveram resultados positivos em relação aos outros experimentos, apenas tendo sido identificadas regras com 56% de sucesso (capaz de identificar 140 das CIs apresentadas); estas, porém, apresentavam pequeno nível de robustez, já que na geração seguinte não foram capazes de manter este nível de sucesso.

### 3.9. Pré-avaliação das CIs

Este esquema utilizado em [Wolz e de Oliveira, 2007] gera, de forma aleatória, os 5% restantes das regras que não sofrem elitismo, porém faz uma rápida pré-avaliação de um conjunto bem maior, e somente então insere as 5% esperadas.

Por exemplo, para uma população de 500 regras, gera-se um conjunto de regras 50 vezes superior aos 5% requeridos (i.e., 25 regras), que no caso são  $50 \times 25 = 1250$  regras; estas são então pré-avaliadas em um conjunto reduzido de CIs geradas aleatoriamente, composto de apenas 10%, do total das 500 envolvidas, ou seja, apenas 50 CIs. Deste processo são então selecionadas as 25 melhores regras, que comporão a população da próxima geração. Observe-se que o objetivo dessa pré-avaliação é criar maiores chances de introduzir na população regras que tenham um nível mínimo de qualidade.

Para este experimento foram utilizadas as variações de comprimentos do reticulado com janelas de tamanho praticamente fixos, com sobreposição de comprimentos, descrito no item anterior.

Os resultados tiveram pouca variação no nível de sucesso já atingido nos experimentos anteriores. Ressalta-se apenas que, algumas execuções apresentaram leve

melhoria levando a níveis de sucesso próximos de 60%, os quais não se mantiveram nas gerações seguintes.

## Capítulo IV: Conclusão

Os experimentos aqui realizados não permitiram identificar uma regra de AC capaz de obter um nível razoável de sucesso na solução do problema da paridade, já que a melhor regra obtida não apresentou eficácia superior a 60%. Ressalte-se que este patamar de qualidade é o mesmo de [de Oliveira e Vaiano, 2005].

Visto que os experimentos conduzidos com a variação do reticulado pouco influenciaram na busca evolutiva, alterou-se o operador de cruzamento aqui empregado (de ponto único), para o proposto em [Franks, 2004] e, conforme visto, ele não trouxe vantagens para o caso em questão. Adicionalmente, apesar da utilização de parâmetros e idéias de sucesso comprovado em [Wolz, de Oliveira, 2007] não se conseguiu aumentar o nível de qualidade anterior

Conclui-se, portanto, que os excelentes resultados obtidos em [Wolz e de Oliveira, 2007] não foram devidos aos métodos de avaliação aqui utilizados, aliados ou não aos valores testados dos parâmetros evolutivos, entre os quais se destaca a altíssima taxa de elitismo de 95%. Tais resultados devem-se, portanto, a outros conceitos importantes lá empregados, principalmente a técnica de compartilhamento de recursos (resource sharing) no processo evolutivo; a arquitetura evolutiva desenvolvida, paralela e em dois níveis; o método de seleção, baseado num mecanismo de ordenação escalonada (scaled ranking); sem contar os próprios recursos computacionais despendidos, muito maiores que aqui.

Mesmo com toda esta dificuldade, a vasta gama de regras que compõem a família de ACs unidimensionais de raio 3, as possibilidades de variação dos métodos de busca através de computação evolutiva, e o sucesso obtido em [Wolz e de Oliveira, 2007] para reticulados ímpares, não podemos concluir que não exista uma regra que solucione o PP

através de computação implícita, nem que boas regras não possam ser encontradas para reticulados pares.

Para trabalhos futuros sugere-se o estudo dos outros conceitos utilizados em [Wolz e de Oliveira, 2007], não abordados nesta pesquisa e que, possivelmente, permitirão identificar o(s) fator(es) que interfere(m) no processo evolutivo e que possibilita(m) identificar boas regras para o problema da paridade.

## Capítulo V: Referências Bibliográficas

[Arbib, 1969] M.A. Arbib. *Theories of Abstract Automata 412p*. Prentice-Hall, 1969.

[Breukelaar and Bäck, 2005] R. Breukelaar and T. Bäck. “Using a genetic algorithm to evolve behavior in multi dimensional cellular automata”. In: H.-G. Beyer, U.-M. O'Reilly, D.V. Arnold, W. Banzhaf, C. Blum, E.W. Bonabeau, E. Cantú Paz, D. Dasgupta, K. Deb, J.A. Foster, E.D. de Jong, H. Lipson, X. Llorca, S. Mancoridis, M. Pelikan, G.R. Raidl, T. Soule, A. Tyrrell, J.-P. Watson and E. Zitzler (eds.). *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO-2005*, New York: ACM Press, V.1, 107-114, 2005.

[de Oliveira, Bortot e Oliveira, 2006] P.P.B. de Oliveira, J.C. Bortot e G.M.B. Oliveira. “The best currently known class of dynamically equivalent cellular automata rules for density classification”. *Neurocomputing*, 70(1-3):35-43, 2006.

[de Oliveira e Vaiano, 2005] P.P.B. de Oliveira e R.B. Vaiano. “Buscando um autômato celular que resolva o problema da paridade”. In: A.C.B. Garcia e F.S. Osório, eds. *Anais do 5º Encontro Nacional de Inteligência Artificial, Sociedade Brasileira de Computação*, CD-ROM, p.1202-1205, 2005.

[Franks, 2004] A. Franks. “Understanding Evolved Strategies for System-Wide Coordination in Noisy Environments”, *Intel Science Talent Search Papers*, November 17, 2004.

[Fukús, 1997] H. Fukús. “Solution of the density classification problem with two cellular automata rules”. *Physics Review E*, 55:2081R-2084R, 1997.

[Silva e de Oliveira, 2006] P.S. Germano da Silva e P.P.B. de Oliveira, “Variações nas configurações iniciais de autômatos celulares unidimensionais na busca evolutiva por

- regras para o problema da paridade”, *Boletim Técnico de Pós-Graduação* 079/2006 - Universidade Presbiteriana Mackenzie, 2006.
- [Silva e de Oliveira, 2007] P.S. Germano da Silva e P.P.B. de Oliveira. “Evolutionary Search for Cellular Automata in the Parity Problem: Variations in Rule Evaluation Methods”. Submetido ao *13<sup>th</sup> Portuguese Conference on Artificial Intelligence - EPIA* 2007.
- [Hillis, 1992] W.D. Hillis *Co-evolving parasites improve simulated evolution as an optimization*. In C.G. Langton, C.Taylor, J.D. Farmer, and S.Rasmussen, eds., *Artificial Life II* Addison-Wesley.
- [Holland, 1975] J.H. Holland. *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, 1975.
- [Hopcroft, Ullman e Motwani, 2005] John E. Hopcroft, Rajeev Motwani e Jeffrey D. Ullman, *Introdução à teoria dos autômatos, linguagens e computação* - 2ª edição, Editora Campus, 2002. <http://www-db.stanford.edu/~ullman/ialc.html>
- [Jacob, 2001] Christian Jacob. *Illustrating Evolutionary Computation with Mathematica*, Academic Press, 2001.
- [Juillé e Pollack, 1998] Juillé, H.; Pollack, Jordan B. “Coevolving the “Ideal” trainer: application to the discovery of cellular automata rules”. In: J.R. Koza, W. Banzhaf, K. Chellapilla, M. Dorigo, D.B. Fogel, M.H. Garzon, D.E. Goldberg, H. Iba, and R.L. Riolo (eds.), *Genetic Programming 1998: Proceedings of the Third Annual Conference: San Francisco, CA: Morgan Kaufmann*, 1998.
- [Koza, 1993] Koza, J.R.. *Genetic Programming*, The MIT Press, 1993.
- [Lee, Xu e Chau, 2001] K.M. Lee, H. Xu e H.F. Chau. “Parity Problem with a Cellular Automaton Solution”, *Physical Review E*, 64:026702/2-026702/4, 2001.

- [Mitchell, 1998] Melanie Mitchell. *An Introduction to Genetic Algorithms*, MIT Press, Cambridge, MA, 1998.
- [Mitchell, 1996] Melanie Mitchell. "Computation in Cellular Automata: A Select Review". In: *Nonstandard Computation*. Weinheim: VCH Verlagsgesellschaft, 1996.
- [Mitchell, 1993] Melanie; Hrabar, P. T.; Crutchfield, J. P. "Evolving cellular automata to perform computations: mechanism and impediments". *Physica D*, 75:361-391, 1993.
- [Moreira, Mathur, Diemeier e Amaral, 2004] A.A. Moreira, A. Arthur, D. Diemeier e L.A.N Amaral. "Efficient System-Wide Coordination in Noisy Environments", *PNAS*, 101(33): 12085-12090, 2004.
- [Myhill, 1964] J. Myhill. "The Abstract Theory of Self-Reproduction". In: M.D. Mesarovic, editor. *Views on General Systems Theory*. New York: John Wiley & Sons Inc, p.106-118, 1964.
- [Oliveira, 2003] G.M.B. Oliveira. "Autômatos Celulares: aspectos dinâmicos e computacionais". Mini-cursos de inteligência artificial, *Anais do XXIII Congresso da Sociedade Brasileira de Computação*, CD-ROM, Campinas, 2003.
- [Papadimitriou, 2000] Harry R. Lewis E Christos H. Papadimitriou. *Elementos de Teoria da Computação*, Bookman, 2000
- [Sipper 1998] Moshe Sipper. "Computing with cellular automata: Three cases for nonuniformity". *Physical Review E*, 57(3):3589-3592, 1998.
- [Sipser, 1997] Michael Sipser. *Introduction to the Theory of Computation*, PWS Publishing Company, 1997.
- [Toffoli, 1987] T. Toffoli e N. Margolus. *Cellular Automata Machines: A New Environment for Modeling*, The MIT Press, Cambridge MA, 1987.

- [Von Neumann, 1966] John Von Neumann In: A.W. Burks (ed). *Theory of Self-Reproducing Automata*, University of Illinois Press, Illinois, 1966.
- [Wolfram, 2002] S. Wolfram. *A New Kind of Science*, Wolfram Media, 2002.
- [Wolfram, 2003] S. Wolfram. *The Mathematica Book, 5th Edition*, Wolfram Media, 2003.
- [Wolfram, 1984] S. Wolfram. “Universality and complexity in cellular automata”, *Physica D*, 10:1-35, 1984. Disponível na Internet em: <http://www.stephenwolfram.com/publications/articles/ca/84-universality/index.html>
- [Wolz e de Oliveira, 2007] D. Wolz e P.P.B. de Oliveira. “Very Effective Evolutionary Techniques for Searching Cellular Automata Rules Spaces”, *Submetido ao Jornal of Cellular Automata*, 2007.

# Livros Grátis

( <http://www.livrosgratis.com.br> )

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)  
[Baixar livros de Literatura de Cordel](#)  
[Baixar livros de Literatura Infantil](#)  
[Baixar livros de Matemática](#)  
[Baixar livros de Medicina](#)  
[Baixar livros de Medicina Veterinária](#)  
[Baixar livros de Meio Ambiente](#)  
[Baixar livros de Meteorologia](#)  
[Baixar Monografias e TCC](#)  
[Baixar livros Multidisciplinar](#)  
[Baixar livros de Música](#)  
[Baixar livros de Psicologia](#)  
[Baixar livros de Química](#)  
[Baixar livros de Saúde Coletiva](#)  
[Baixar livros de Serviço Social](#)  
[Baixar livros de Sociologia](#)  
[Baixar livros de Teologia](#)  
[Baixar livros de Trabalho](#)  
[Baixar livros de Turismo](#)