

AUTOMATO CELULAR GÊNESIS: Um Ambiente Virtual Baseado Em Autômato Celular Para Simulação de Vida Artificial

Dr. Alexandre Maniçoba De Oliveira

Laboratório Maxwell - IFSP, Cubatão, SP, Brasil

Dr. João Francisco Justo

Departamento de Engenharia de Sistemas Eletrônicos - EPUSP, São Paulo, SP, Brasil

Dr. Carlos Henrique Barroqueiro

Laboratório Maxwell - IFSP, Cubatão, SP, Brasil

Me. Arnaldo de Carvalho Júnior

Laboratório Maxwell - IFSP, Cubatão, SP, Brasil

Resumo: Este trabalho apresenta um modelo teórico e conceitual de um ambiente virtual para a observação da Vida Artificial (VA) pelo uso de Autômatos Celulares (AC) com o intuito de simular a interação entre os agentes a partir de regras simples, reproduzindo “*in silício*” o ciclo básico da vida: nascer, amadurecer, alimentar-se, reproduzir e evoluir, envelhecer e morrer, com base em comportamentos naturais de organismos biológicos. Neste ambiente os organismos virtuais (agentes) são dinâmicos e têm seu comportamento regido por processamento local, simples e distribuído, com base nos modelos presa-predador de Lotka e Volterra e no AC jogo da vida de John Conway e inspirado nos trabalhos de VA com AC de Von Neumann, Wolfram e Langton. O modelo proposto simula condições em que os agentes de diferentes espécies disputam espaço na grade, onde a espécie maior se alimenta das espécies menores. O programa de VA será “*vivido*” em uma grade bidimensional formando um espaço do tipo “*vitrífico*” da lâmina artificial (analogia as placas de Petri). Os resultados das simulações da VA do AC proposto apresentam inicialmente um comportamento caótico com o surgimento de atratores de ciclo fechado após algumas dezenas de iterações e que persistem mesmo após dez mil iterações, evidenciando o funcionamento complexo do modelo VA proposto mesmo baseado em um conjunto de regras básicas, atendendo as definições para este tipo de modelo segundo a literatura.

Palavras-chave: Vida Artificial. Autômato Celular. Inteligência Artificial. Preza e Caçador.

Abstract: This investigation presents a theoretical and conceptual model of a virtual environment for the observation of Artificial Life (AL) using Cellular Automata (CA), aiming to simulate the interaction between the agents from simple rules, reproducing

"in silicon" the elementary cycle of life: to born, mature, feed, reproduce, evolve, grow old, and die, based on the natural behaviors of biological organisms. In this environment, virtual organisms (agents) are dynamic and have their behavior governed by simple, distributed local processing, based on the prey-predator models of Lotka and Volterra and in the CA game of John Conway's life and inspired by AL works with CA by von Neumann, Wolfram, and Langton. The proposed model simulates conditions in which agents of different species compete for space in the grid, where the larger species feeds on smaller species. The AL program will be "lived" in a two-dimensional grid forming a "vitrified" type of artificial blade (analogy to Petri dishes). The results of the simulations of the AL of the proposed CA initially present a chaotic behavior with the appearance of closed-loop attractors after a few dozen iterations and that persist even after ten thousand iterations, evidencing the complex functioning of the proposed AL model even based on a set of rules, taking into account the definitions for this type of model, according to the literature.

Keywords: *Artificial Life. Cellular Automata. Artificial Intelligence. Food and hunter, game of life.*

INTRODUÇÃO

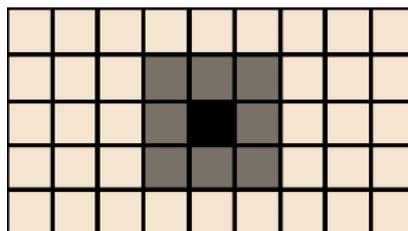
No contexto geral do estudo de VA, observou-se haver várias ferramentas de simulação computacional para seu entendimento. A aplicação para este tipo de modelo se destina a tarefas complexas que exigem processamento distribuído e altamente paralelo como na proposta da utilização de métodos de AC para a otimização de funções não lineares (KENNEDY; EBERHART, 1995). Desta forma, buscou-se na natureza a inspiração biológica para o modelo proposto a partir da laica observação do comportamento organizacional de micro-organismos em placas de Petri. Já na área computacional, a inspiração partiu dos modelos de AC presa-predador (KAPLAN ; GLASS, 1995) e o Jogo da Vida (CONWAY, 1970).

O modelo conceitual proposto simula a interação de uma população formada por agentes de diferentes espécies, para que seja perceptível a evolução, manutenção e até mesmo a extinção de espécies com base em regras simples de interação de organismos naturais originadas da observação da "vida como ela é" de forma a propor um ambiente propício ao estudo da "vida como ela poderia ser"

(LANGTON, 1986; WOLFRAM, 1998). Para a realização deste modelo teórico de simulação de VA, foi necessário antes mesmo de criá-lo, entender algumas definições para “o que é a vida” de forma a possibilitar o ato criador (Gênesis) de vida em outro meio, que não seja o natural (baseado em cadeias de carbono), mas sim idealizado por meio de um ambiente virtual onde os detalhes de implementação da vida são resumidos em um modelo particular tão parecido com a vida que transpasse a barreira própria do modelo tornando-se exemplos de vida “in silício” [6]. Para isso, esse ambiente de VA baseado em AC foi criado fisicamente na memória RAM e estruturado logicamente como uma matriz ou grade bidimensional (2D), denominada apenas como “**D**”, na forma toroidal, ou seja, não possui região de contorno definida pois seus limites (bordas) são periódicos na forma de uma extensão da grade onde cada célula da borda tenha, assim como as células no interior da grade, oito células vizinhas. Nesta grade, cada posição de memória, doravante denominada apenas como automata ou célula, é identicamente programada de forma a interagir, entre si, em função de um conjunto finito de regras, que alteram seus estados de forma paralela e sincronizada (WOLFRAM, 1998; CHATE e MANNEVILLE, 1992) que serão tratadas neste documento como regras de transição.

Para criar o modelo proposto foi utilizado o método de modelagem dos agentes baseado em autômatos celulares onde cada automata será baseado em simples regras em função do estado próprio e em relação aos agentes em sua vizinhança de Moore (VM), ilustrada na Figura 1.

Figura 1 – Vizinhança de Moore, onde ao centro encontra-se o agente na cor preta, em seu entorno, representadas pela cor cinza, estão as células visíveis que interferem diretamente nas regras de transição do agente e as demais células não influenciam diretamente o agente (as brancas), são para este, invisíveis.



Fonte: (LANGTON, 1986; SYSI-AHO, 2005)

Estas regras são conhecidas como regras de transição (rt) e formam um sistema dinâmico que descrevem o comportamento local e autônomo de cada agente criando a aproximação do conceito de individual (WOLFRAM, 1998). Toda a interação gera o comportamento coletivo e emergente levando a resultados macroscópicos.

O AUTÔMATO CELULAR PARA VIDA ARTIFICIAL E SUA APLICABILIDADE

O início dos estudos teóricos sobre os ACs foram propostos por John Von Neumann (1966). O interesse de estudo sobre AC de Neumann relacionava o comportamento de organismos biológicos com a teoria dos Autômatos, sobretudo no que diz respeito ao fenômeno de auto reprodução. Com base nestas observações, Neumann levantava a seguinte pergunta: Qual tipo de organização lógica seria suficiente para que um autômato tenha a capacidade de se auto reproduzir? (DE AGUIAR; COSTA, 2001). Segundo Shannon (1958), Von Neumann dedicou grande parte de seus últimos anos de vida ao trabalho da teoria de autômatos. Esta teoria foi significativamente importante, mais tarde, durante a Segunda Guerra Mundial e depois, quando desenvolveu pesquisas em computadores eletrônicos de grande porte.

Anos mais tarde, Christopher G. Langton (1986) implementou um modelo lógico de vida a partir da inspiração obtida na observação dos estudos de bioquímica. Este estudo levou-o a concluir que seria possível modelar sistemas computacionais com características de sistemas naturais vivos, dando origem ao conceito que ele batizou, pela primeira vez, de Vida Artificial (VA) ou do inglês (*Artificial Life* ou apenas *ALife*) onde ele, Langton, define-a como: “vida criada pelo Homem em contraste à natureza”.

Seu modelo de AC de VA foi concebido para ser executado de maneira massivamente paralela de forma que muitos elementos computacionais (agentes) pudessem ser simulados com interações simultâneas entre si distributivamente e com características de vizinhanças, ou seja, cada automata estaria conectado apenas com os agentes de sua vizinhança, o que potencializaria o que ele chamou de massivo

paralelismo e com uma arquitetura funcional que satisfizesse os critérios propostos por John Von Neumann [4].

Além de se inspirar nos estudos de AC voltados a auto reprodução de Neumann, Langton investigou o trabalho de físicos (FARMER et al., 1984; FREDLDN et al., 1982; TOFFOLI, 1977) com autômatos celulares usando a última abordagem de Neumann, onde observou que AC são sistemas dinâmicos de espaço e tempo discretos, ou seja, é o sistema onde suas variáveis são alteradas em função dos seus valores atuais. Desta forma, o funcionamento de muitos sistemas dinâmicos é regido por sistemas de equações diferenciais não lineares, o que os tornam sistema complexos e difíceis de analisar (LANGTON, 1986).

Para Langton (1986) o estudo de tais sistemas (dinâmicos) envolve a análise do espaço definido por todas as variáveis (*phase space* - ps). Existem três possibilidades de comportamento, a longo prazo, de um sistema no ps: 1º vai parar de se mover por completo, ficará inanimado; 2º ele vai cair em um ciclo fechado; ou 3º não vai terminar em si mesmo. No primeiro caso, o sistema evoluiu para um ponto limite ou ponto fixo, no segundo caso, o sistema evoluiu para um ciclo de limite, e no terceiro caso o sistema evoluiu para algo chamado um *strange attractor* (atrator estranho, curioso, único, singular, novo). Na verdade, todos os três são tipos de atratores, este conceito deriva do fato de que se um sistema está em um estado que é “quase” um atrator no ps, que geralmente evolui “para” o estado ou o ciclo representado por aquele atrator. Ainda segundo Langton, os sistemas regidos pela terceira classe de atrator, os *strange attractors*, comportam-se de forma caótica, aleatório e imprevisível, desta forma, a menos que seja simulado passo a passo sua evolução no tempo, teremos que recorrer a ferramentas probabilísticas para descrever seu comportamento, mesmo sendo estes sistemas regidos por comportamento totalmente determinístico.

Analogamente, o AC possui o *state space* – ss ou seja, região onde estão contidos os estados atuais de cada automata, desta forma, a qualquer momento há sempre uma distribuição exclusiva dos estados de cada automata e essa distribuição é representada como um ponto no ss (LANGTON, 1986). A evolução, em função do tempo, de qualquer automata celular pode ser estudada através da observação do

“caminho” que resulta no seu ss. Um detalhe interessante é que por vezes, atratores abundam também nos ss de AC, sendo esse um dos principais recurso que chamaram a atenção dos físicos. AC muitas vezes exibem comportamentos característicos das três classes de atratores. Stephen Wolfram realizou um detalhado estudo dos AC e sua afinidade com sistemas dinâmicos (WOLFRAM, 1984, 1985, 1998). Wolfram considerou os padrões obtidos por AC evoluindo de "sementes" simples, formadas com valores diferentes de zero e observou que ao aplicar algumas regras locais (rt) foi possível criar comportamentos simples, em outros casos produziu padrões complexos. Para Wolfram, (1998), quatro padrões distintos foram observados após um extenso estudo empírico: 1º desaparece com o tempo; 2º evolui para um tamanho fixo finito; 3º cresce indefinidamente a uma velocidade fixa; 4º cresce e formatos irregulares.

Ao se realizar um paralelo entre os atratores propostos por Langton (1986) e os padrões propostos por Wolfram (1998), pode-se notar uma certa semelhança de definição entre o 1º atrator e os 1º e 2º padrões, onde o AC converge para um estado fixo e inalterado. Entre o 2º atrator e o 3º padrão, descreve-se um comportamento do AC regido por um ciclo repetitivo e constante e por fim, o que Langton (1986) definiu como atrator estranho (3º) e Wolfram (1998) como padrão irregular (4º), define o estado em que o AC converge para um estado caótico (irregular), imprevisível e aleatório.

Observa-se desta forma que tanto para Langton (1986) quanto para Wolfram (1998) os modelos baseados em AC são tipicamente mais indicados para solução de sistemas físicos em regimes altamente complexos, tais como sistemas químicos e biológicos em que ocorrem limiares discretos.

Atualmente há uma variedade de aplicações para os AC, por exemplo para a modelagem da dinâmica da redistribuição de solo, provocada pela erosão com AC proposta por Heung et. al. (20013) para o projeto de um *framework* gráfico para estudo e simulação em tempo real de modelos de enchentes urbanas causadas por precipitações pluviais (GUIDOLIN, 2011) e/ou para o estudo do crescimento de fluxo turístico na cidade de Sanpo da província de Hebei com base em um simulador espacial e temporal implementado com AC (YANG, 2013).

Como pode ser observado, modelos de AC são prioritariamente aplicáveis a sistemas de equações complexas, e a simulação da VA e toda interação entre indivíduos orgânicos pode ser implementada com sucesso pelo uso de AC como no modelo matemático proposto por Jafelice et. al. (2011) baseado em AC para o estudo, por simulação, em uma abordagem de “a vida como ela poderia ser”, da evolução do HIV intra-organismo dos indivíduos infectados e sob o tratamento antirretroviral.

Desta forma observa-se que AC’s são modelos artificiais úteis para dar origem a padrões emergentes complexos com base em regras relativamente simples, entretanto vale salientar que AC’s convencionais não possuem nenhum tipo de realimentação evolucionária baseado em regras genéticas, ou seja, a cada nova geração, nenhuma marca indelével é herdada da geração anterior (PAVLIC, 2014). Assim sendo, uma das características propostas no algoritmo Gênesis, apresentado neste trabalho, é a criação de um AC capaz de sintetizar VA com base em agentes que possam realizar completamente o ciclo de vida: nascer, amadurecer, alimentar-se, reproduzir, envelhecer, morrer e EVOLUIR.

Para isso, foram criadas rt e atributos que promovem o ato criador do algoritmo Gênesis que será apresentado nas próximas Seções.

FORMALIZAÇÃO DO AC GÊNESIS

Com base na revisão bibliográfica de reconhecidas práticas para o estudo e desenvolvimento de VA com o uso de AC, buscou-se como resultado deste trabalho a implementação de um AC de VA na forma de um sistema dinâmico, determinístico e discreto no espaço e tempo, operando uniforme e regularmente em **D** e alterando os seus estados através de interações locais para o estudo de VA (ADAMATZKY; KOMOSINSKI, 2005).

O sistema dinâmico com base em **D** foi formado por várias células (**c**) e caso o estado de **c** (**S_c**) for maior que zero, afirma-se que **c** é uma região em **D** habitada por um agente “vivo” (**c=a**), onde:

$$D=\{c_1, c_2, \dots, c_n / c=a \text{ sse } (S_c > 0) \quad Z^+\}, \text{ onde:}$$

n - é o número de células de D .

A VM ou campo visual de a , denominado Va , que tem o diâmetro r de um agente a , é todo o intervalo representado a seguir:

$$Va^i_{x,y} = \{c_{(x+r,y+r)} / 0 \leq |r|, r \in \mathbb{Z}\}, \text{ onde:}$$

i – identificador do agente;

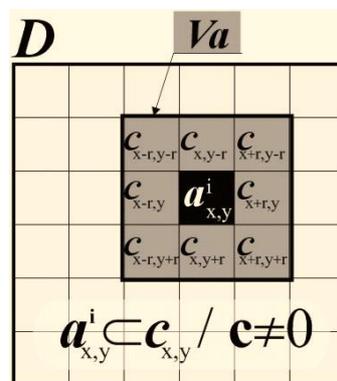
x,y – coordenada de a em D ;

r – diâmetro de $D = \{-1,0,1\}$;

Cada célula de D deverá assumir um estado entre um conjunto de estados diferentes. No modelo proposto cada c de D será representado por um vetor de estados $S=(S_0,S_1)$ tal que a parte inteira de S_0 representará a espécie (de 0 a ES por exemplo), a parte centesimal o **gene** de S_0 (de 0 a 0,999) e S_1 a saúde do agente (de 0 a $S_{1max} = 9$ por exemplo). Todas as c não nulas ($c \neq 0$) que possuem o estado $S_0 \neq 0$ e $S_1 \neq 0$, são consideradas um a , tornando-se passíveis da aplicação das **rt**. Com isto tem-se um ganho considerável no desempenho do sistema, visto que as c nulas ($c=0$) que possuem o estado $S_0=0$ ou $S_1=0$, não necessitaram de processamento.

Segue a arquitetura de implementação da Va como ilustra a Figura 2.

Figura 2 – Representação hipotética de D contendo a e sua Va



Fonte: autoria própria

Formalmente o AC Gênesis é definido através da quintupla:

$a = (S, S_i, V_a, d, f)$, onde:

S - é um conjunto finito de estados;

S_i - são os estados iniciais;

d- dimensão de **D** \mathbb{Z}^+ ;

f - conjunto de **rt**;

Após a implementação deste AC de VA, foi possível criar um ambiente “*in silício*” de VA onde são observadas diversas possibilidades de interações entre os agentes ao longo de várias épocas, o que foi possível contribuir para a observação acadêmica de tais técnicas de processamento massivamente paralelo e localizado e seus resultados globais e macroscópicos, emergindo da aplicação de **rt**'s simples e individualmente em cada agente. Este comportamento, bem como o algoritmo e as suas regras são apresentados a seguir.

O AUTÔMATOS CELULAR GÊNESIS

A partir do estudo das investigações e teorias de AC propostas por Neumann, com base nos modelos presa-predador de Lotka e Voltera e no AC jogo da vida de John Conway e inspirado nos trabalhos de VA baseado em AC de Wolfram e Langton, é proposto este algoritmo que por se tratar de um AC para reproduzir VA, foi chamado de Gênesis, em referência lúdica ao ato criador apresentado pelo primeiro livro do Pentateuco, com o mesmo nome.

Este tem por conceito fundamental não a busca por reproduzir sistemas complexos baseado em complicadas equações diferenciais, mas sim o desenvolvimento de um ambiente de simulação sistêmica de interações celulares (**c₁**, **c₂**, **c₃**, ..., **c_n**) com regras simples. Desta forma, o foco deste algoritmo não é representar um intrincado sistema com equações complexas, mas permitir que toda a complexidade possa emergir através da interação de cada automata (agente) em função de suas simples regras de transição (WOLFRAM, 1998).

Seu funcionamento será apresentado nas próximas Subseções.

A. Descrição geral do AC Gênesis

Inspirado no comportamento do sistema auto organizado de colônias de micro-organismos como bactérias e/ou protozoários, mas sem a pretensão de reproduzir integralmente tais comportamentos, mas sim em propor um estudo dos AC's de VA, desenvolve-se um AC de forma a simular um ambiente virtual **D** podendo ser habitado por n agentes **a**.

As regras básicas (**rt**) para o AC Gênesis são: alimentação, reprodução, movimentação, nascimento, morte e evolução.

Vale salientar que para garantir a manutenção mínima da vida no AC, os agentes com $S_0 = 1$ surgem espontânea e aleatoriamente em **D**, por uma **rt**.

Na sequência são apresentadas as demais **rt**'s que dão “vida” ao AC proposto, a começar pela capacidade do agente em se movimentar.

B. Da movimentação

No instante em que o agente a_1 se move, entende-se que o mesmo gaste energia para isso, desta forma estado S_{1a_1} é decrementado em 5% (valor padrão $dS_1 = 0.05$) de S_{1max} . Tal **rt** pode ser observada na linha 22 do pseudocódigo da Tabela 1.

A partir de qualquer posição do espaço do **grid D**, o agente a_1 poderá “passear” livremente em sua VM de diâmetro r a cada iteração, caso haja uma **c** vazia e ele a “*enxergue*”. Para isso a agente escolhe aleatoriamente a célula de sua **Va** e verifica se a mesma está vazia $c=0$ ou se é ocupada por outro agente a_2 . Caso esteja vazia move-se para ela em $t+1$. Caso esteja ocupada $c=a_2$, não se move e três possíveis situações ocorrerão:

1ª se a espécie de a_2 for menor que a_1 , a regra de alimentação poderá ser utilizada e é finalizada a transição de a_1 .

2ª se a espécie de a_2 for igual a a_1 , e ambos estejam em “idade” fértil, ($S_{1a1} > trepro * S_{1max}$ and $S_{1a2} > trepro * S_{1max}$), a regra de reprodução será utilizada, sendo finalizada a transição de a_1 com o possível surgimento de um filho, neste caso o agente a_3 . Caso **contrário**, a regra é finalizada.

3ª se a espécie de a_2 for maior que a de a_1 , a regra é finalizada, ou seja, a_1 desiste de se mover em direção de a_2 , caso contrário o “bicho” pega literalmente.

Tabela 1 - Pseudocódigo da *rt mover*

Nº	Pseudocódigo por linha
01	If integer ($S_{0xy} > 0$) and ($S_{1xy} > 0$) then
02	$x_2 := x - 1 + (RND(1) * 3 * r)$;
03	$y_2 := y - 1 + (RND(1) * 3 * r)$;
04	If ($x_2 > CL$) then
05	$x_2 := 1$;
06	End If
07	If ($y_2 > RW$) then
08	$y_2 := 1$;
09	End If
10	If ($x_2 < 1$) then
11	$x_2 := CL$;
12	End If
13	If ($y_2 < 1$) then
14	$y_2 := RW$;
15	End If
16	If integer ($S_{0x2y2} = 0$) or integer ($S_{1x2y2} = 0$) then
17	$S_{0x2y2} := S_{0xy}$;
18	$S_{1x2y2} := S_{1xy}$;
19	$S_{0xy} := 0$;
20	$S_{1xy} := 0$;
21	End If
22	$S_{1xy} := S_{1xy} - S_{1max} * dS1$;
23	End If

C. Da alimentação

Cada agente poderá se alimentar dos agentes de espécie inferior e por sua vez, se tornar alimento de agentes de espécies superiores. No caso em que SO_{a1} for maior que SO_{a2} , o agente a_1 poderá assumir a automata c de a_2 , ou seja, c de a_1 é copiada para a c de a_2 , em seguida a c de a_1 é zerada, se e somente o resultado da

diferença entre o valor da maior espécie (ES) por um número sorteado aleatoriamente entre 0 e ES-1 for menor que a vontade de alimentar-se, configurável pelo parâmetro **fome** (linha 9 na Tabela 2). Neste caso, o agente **a₂** deixa de existir e será realizado um incremento da vida de **a₁** (**S_{1a1}**) em um valor equivalente ao produto de **S_{1max}**, **dS1** e o parâmetro **nutrição**, garantindo a manutenção da vida de **a₁** que se alimentou de **a₂**. Os agentes de espécie igual a 1 alimentam-se das células nulas ou zeradas, simbolizando material orgânico inanimado.

A **rt** pode ser observada na forma de pseudocódigo na Tabela 2.

Tabela 2 - Pseudocódigo da *rt* alimentar

Nº	Pseudocódigo por linha
01	<i>If integer(S_{0x2y2}) < integer (S_{0xy}) and (ES-RND(ES))<fome then</i>
02	<i>S_{0x2y2} := S_{0xy};</i>
03	<i>S_{1x2y2} := S_{1xy} + (S_{0max} * dS1*nutricao);</i>
04	<i>S_{0xy} := 0;</i>
05	<i>S_{1xy} := 0;</i>
06	<i>End If</i>
07	<i>If integer (S_{0x2y2}) <1 and integer (S_{0xy}) = 1 then</i>
08	<i>S_{0x2y2} := S_{0xy};</i>
09	<i>S_{1x2y2} := S_{1xy} + (S_{0max} * dS1*nutricao);</i>
10	<i>S_{0xy} := 0;</i>
11	<i>S_{1xy} := 0;</i>
12	<i>End If</i>
13	<i>End If</i>

D. Da reprodução

Cada agente poderá se reproduzir desde que esteja em “idade fértil”, ou seja, **S_{1a1}** deverá ser maior que o produto de **S_{1max}** pelo parâmetro **trepro** e encontre um parceiro de mesma espécie, em sua vizinhança (**Va₁**), também em “idade fértil”.

O se houver espaço livre na vizinhança **Va₁**, um neo-agente **a₃** surge em uma **c=0** de **Va₁** com o estado de **Soa₃** igual a soma da média aritmética de **Soa₁** e **Soa₂** com uma evolução de 25% de seu gene (valor padrão contido no parâmetro **tevol**). Neste instante o neo-agente **a₃** sofre uma evolução em relação a seus pais, podendo inclusive mudar para uma espécie superior.

A procura por espaço livre na reprodução é feita a partir de um número de n tentativas randômicas onde, no qual n equivale a diferença do número de espécies (ES) pela espécie do agente em transição, ou seja, os agentes de menor espécie terão mais tentativas para encontrar uma c nula em Va do que os de maior espécie, garantindo-lhes maiores chances de reprodução.

A rt reproduzir pode ser observada como pseudocódigo na Tabela 3.

Tabela 3 - Pseudocódigo da rt reproduzir

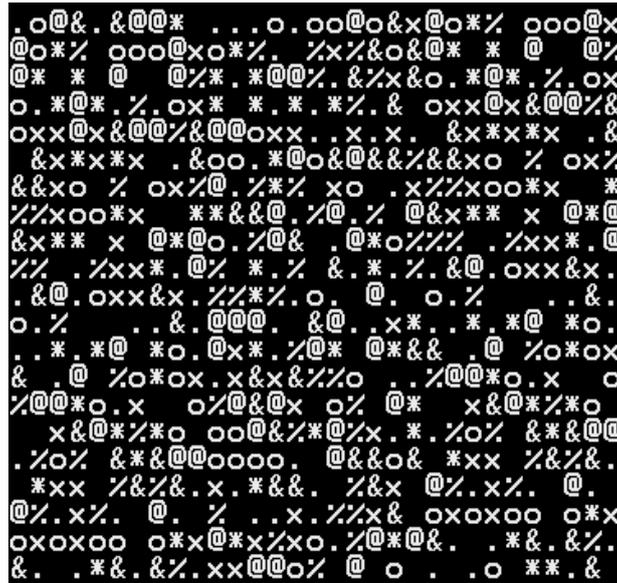
N o	Pseudocódigo por linha
01	<i>If integer</i> (S_{0x2y2}) = <i>integer</i> (S_{0xy}) <i>then</i>
02	<i>If</i> $S_{1xy} > (S_{1max} * trepro)$ <i>and</i> $S_{1x2y2} > (S_{1max} * trepro)$ <i>then</i>
03	<i>For</i> $k = 0$ <i>to</i> <i>integer</i> ($ES - S_{0xy}$)
04	$x3 := x - 1 + (RND(I) * 3 * r);$
05	$y3 := y - 1 + (RND(I) * 3 * r);$
06	<i>If</i> $x3 > CL$ <i>then</i>
07	$x3 := 0;$
08	<i>End If</i>
09	<i>If</i> $x3 < 0$ <i>then</i>
10	$x3 := CL;$
11	<i>End If</i>
12	<i>If</i> $y3 > RW$ <i>then</i>
13	$x3 := 0;$
14	<i>End If</i>
15	<i>If</i> $y3 < 0$ <i>then</i>
16	$y3 := RW;$
17	<i>End If</i>
18	<i>If</i> $S_{0x3y3} < 1$ <i>then</i>
19	$S_{0x3y3} := (S_{0xy} + S_{0x2y2}) / 2 + tevol;$
20	$S_{1x3y3} := S_{1max};$
21	<i>End If</i>
22	<i>Next</i>
23	<i>End If</i>
24	<i>End If</i>

RESULTADOS DA EXECUÇÃO DO ALGORITMO GÊNESIS

Nesta Seção serão apresentados e comentados alguns resultados obtidos após realizadas dez mil iterações do algoritmo Gênesis com seus valores de

configuração todos como padrão o que garante o estado inicial como ilustrado na Figura 3.

Figura 3 – Ilustração da matriz D com uma população de agentes de diferentes espécies no estado inicial



Fonte: autoria própria

Neste caso o parâmetro ES que determina o número de espécies, está ajustado para 8, desta forma as possíveis espécies são {0, 1, 2, 3, 4, 5, 6, 7} e são representados pelos sete primeiros caracteres da legenda apresentada na Figura 4.

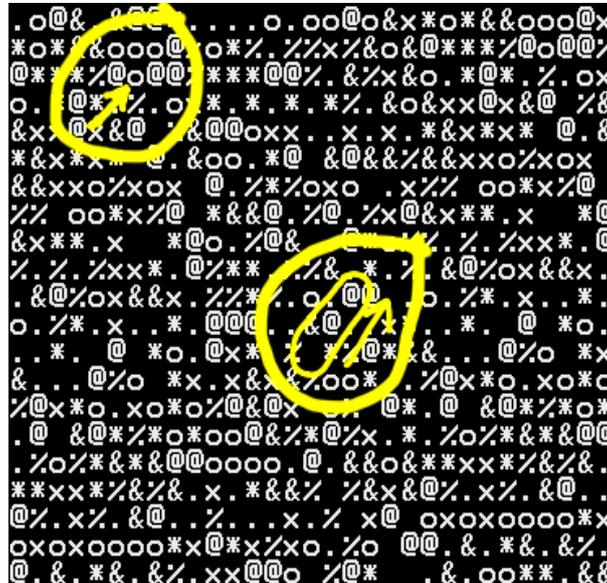
Figura 4 – Ilustração da legenda de nove diferentes espécies de agentes sendo representada da menor para a maior da esquerda para a direita



Fonte: autoria própria

Após a 1ª iteração pode ser observado o surgimento de um novo agente da espécie 3 (o) surgindo da *rt* reproduzir e que pode ser observado no canto superior esquerdo da matriz **D** como ilustra o detalhe da Figura 5.

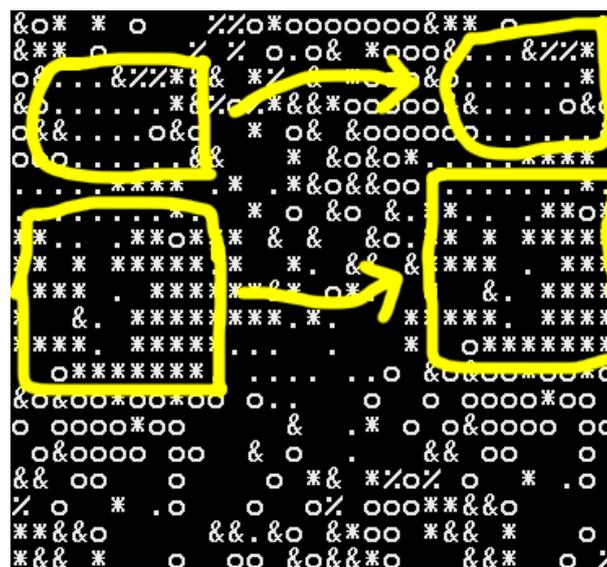
Figura 5 – Ilustração da matriz D com uma população de agentes de diferentes espécies após a 1ª iteração com o detalhe na parte superior esquerda de uma reprodução e na parte central uma movimentação



Fonte: autoria própria

Ainda na Figura 6 pode ser observada a aplicação da *rt mover* em dois agentes da espécie 6 (@), que movem-se juntos no sentido diagonal para cima.

Figura 6 – Ilustração da matriz D após a 100ª iteração onde pode ser observada o aparecimento de atratores de ciclo fechado



Fonte: autoria própria

Após algumas dezenas de iterações observou-se o surgimento de padrões repetitivos ao longo da matriz, como se houvesse um tipo de comportamento coletivo entre os agentes formando algo como um macro sincronismo ou um organismo maior formado pelo conjunto de **a's**, mesmo quando a **Va** se limita a uma distância euclidiana bem menor do que o comprimento da matriz **D** o que evidencia um comportamento conjunto que emerge do ambiente vivido pelos agentes, como ilustra a Figura 7. Este comportamento é denominado por Langton [4] como atratores cíclicos fechados e por Wolfram [5] como padrão de evolução para um tamanho fixo.7

Figura 7 – Ilustração da matriz D após a 10000ª iteração onde pode ser observada a predominância de atratores de ciclo fechado mesmo após milhares de gerações



Fonte: autoria própria

Este comportamento sincronizado, e/ou coletivo, que emerge do complexo “interagir dos vários agentes” perpetua-se mesmo após milhares de épocas e pode ser observado na Fig. 7 que ilustra o aparecimento de padrões repetitivos após dez mil (10.000) iterações.

Na próxima Seção serão apresentadas as considerações finais sobre a análise dos resultados do funcionamento do algoritmo Gênesis.

CONCLUSÕES

Este trabalho apresentou um modelo teórico e conceitual de ferramenta algorítmica baseada em Automatos Celulares para simulação de Vida Artificial. Inicialmente foi realizado uma ampla revisão bibliográfica de trabalhos de pesquisadores que iniciaram os estudos de AC e VA, sendo os principais Von Neumann, Langton e Wolfram e após feita a revisão, observou-se que um AC para simular a VA é um sistema dinâmico de espaço e tempo discreto, onde suas variáveis são modificadas em função de seus valores atuais, sendo assim, observou-se que ao contrário da maioria dos sistemas dinâmicos que são regidos por equações diferenciais não lineares o que os torna complicados, o AC é um ambiente de simulação sistêmica de interações celulares com regras simples que permite que toda a complexidade possa emergir através da interação de cada agente.

As regras que foram estabelecidas aos agentes são chamadas de regras de transição e estabelecem condições para que cada agente possa desta forma realizar o ciclo da vida que se resume em nascer, amadurecer, alimentar, reproduzir, evoluir, envelhecer e morrer.

Um diferencial do modelo conceitual de AC proposto é que em comparação aos AC convencionais, que não possuem nenhum tipo de realimentação evolucionária baseado em regras genéticas, o AC Gênesis é capaz de sintetizar VA que garanta aos agentes a realização do ciclo completo de vida, inclusive quanto à evolução.

O AC Gênesis foi implementado em C++ no ambiente CodeBlocks e seus testes apresentaram os agentes sofrendo modificações em função das regras de transição a cada iteração e observou-se que após algumas dezenas de iterações surgiram padrões repetitivos ao longo da matriz D, evidenciando um comportamento coletivo entre os agentes fruto de atratores que Langton definiu como ciclos fechados. Mesmo após dez mil iterações, estes atratores são comuns e o funcionamento do algoritmo apresentou resultados como os esperados e descritos na literatura.

REFERÊNCIAS

ADAMATZKY, Andrew; KOMOSINSKI, Maciej. **Artificial life models in software**. Heidelberg: Springer, 2005.

CHATÉ, Hugues; MANNEVILLE, Paul. Collective behaviors in spatially extended systems with local interactions and synchronous updating. **Progress of theoretical physics**, v. 87, n. 1, p. 1-60, 1992.

CONWAY, John. The game of life. **Scientific American**, v. 223, n. 4, p. 4, 1970.

DE AGUIAR, Marilton Sanhotene; DA ROCHA COSTA, Antônio Carlos. **Autômatos celulares para análise da monotonicidade da declividade de áreas geológicas**. 2001.

FARMER, Doyne; TOFFOLI, Tommaso; WOLFRAM, Stephen (Ed.). Cellular Automata: **Proceedings of an Interdisciplinary Workshop**, Los Alamos, New Mexico 87545, USA, March 7-11, 1983. North Holland, 1984.

FREDLON, E., LANDAUER, R. and TOFFOLI, T. eds., Proceedings of a Conference on the Physics of Computation, **Int. Journal of Theor. Physics**, v. 21, n. 3-4, 6-7 and 12, 1982.

GUIDOLIN, Michele et al. Design of a graphical framework for simple prototyping of pluvial flooding cellular automata algorithms. 2011.

HEUNG, Brandon et al. Modelling the dynamics of soil redistribution induced by sheet erosion using the Universal Soil Loss Equation and cellular automata. **Geoderma**, v. 202, p. 112-125, 2013.

JAFELICE, R. et al. Dinâmica do HIV com retardo fuzzy baseado em autômato celular. **Biomatemática**, v. 21, p. 177-192, 2011.

KAPLAN, Daniel, and GLASS, Leon. **Understanding nonlinear dynamics**. Vol. 19. Springer, 1995.

KENNEDY, James, and EBERHART, Russell. "Particle swarm optimization." **Proceedings of IEEE international conference on neural networks**. Vol. 4. No. 2. 1995.

LANGTON, Christopher G. Studying artificial life with cellular automata. **Physica D: Nonlinear Phenomena**, v. 22, n. 1, p. 120-149, 1986.

NEUMANN, J. Von. In: BURKS A.W. (Ed.). **Theory of Self-Reproducing automata**. Urbana: University of Illinois Press, 1966.

PAVLIC, Theodore P. et al. Self-referencing cellular automata: A model of the evolution of information control in biological systems. **arXiv preprint arXiv:1405.4070**, 2014.

SHANNON, Claude E. Von Neumann's contributions to automata theory. **Bulletin of**

the **American Mathematical Society**, v. 64, n. 3, p. 123-129, 1958.

SYSI-AHO, Marko et al. Spatial snowdrift game with myopic agents. **The European Physical Journal B-Condensed Matter and Complex Systems**, v. 44, n. 1, p. 129-135, 2005.

TOFFOLI, Tommaso. **Cellular automata mechanics**. PhD (Thesis) - Logic of Computers Group, University of Michigan, 1977.

WOLFRAM, Stephan. Cellular automata as models of complexity. *Nonlinear Physics for Beginners: Fractals, Chaos, Solitons, Pattern Formation, Cellular Automata, Complex Systems*, v. 311, p. 197, 1998.

WOLFRAM, Stephen. Universality and complexity in cellular automata. **Physica D: Nonlinear Phenomena**, v. 10, n. 1, p. 1-35, 1984.

WOLFRAM, Stephen. Twenty problems in the theory of cellular automata. **Physica Scripta**, v. 1985, n. T9, p. 170, 1985.

YANG, Jun et al. Spatiotemporal Simulation of Tourist Town Growth Based on the Cellular Automata Model: The Case of Sanpo Town in Hebei Province. In: **Abstract and Applied Analysis**. Hindawi Publishing Corporation, 2013.