

BUSCA EVOLUTIVA POR ESQUEMAS DE ATUALIZAÇÃO ASSÍNCRONA EM AUTÔMATOS CELULARES PARA SOLUÇÃO DA TAREFA DE CLASSIFICAÇÃO DE DENSIDADE

Osniel Lopes Teixeira (IC) e Eurico Luiz Prospero Ruivo (Orientador)

Apoio: PIBIC Mackpesquisa

RESUMO

Desde seu aparecimento em meados do século passado como objeto de interesse científico, os autômatos celulares têm sido estudados tanto como objetos matemático-computacionais quanto como para a simulação de sistemas complexos reais, como, por exemplo, propagação de doenças, dinâmica de fluídos e modelos de crescimento urbano. Apesar de alguns autômatos celulares apresentarem computabilidade universal, a codificação do problema e leitura da saída produzida por regras de transição demandam acesso global ao conjunto das células, denominado configuração. A Tarefa de Classificação de Densidade (TCD) é um problema clássico e utilizado como referência no estudo de tarefas globais que podem ser realizadas de maneira distribuída. Algoritmos evolutivos têm sido usados de maneira recorrente para buscar em espaços de regras, possíveis candidatos a soluções parciais para o problema. Este projeto utiliza um algoritmo evolutivo para realizar busca no espaço de esquemas de atualização resultando em soluções melhores para aquelas já encontradas na literatura para o problema da TCD.

Palavras-chave: Autômatos celulares, algoritmos genéticos, classificação da densidade.

ABSTRACT

Since its birth in the middle of the last century as an object of scientific interest, cellular automata have been studied both as mathematical and computational objects as well as means to simulate complex systems, such as disease propagation, fluid dynamics and urban growth models. Although some cellular automata show universal computability, it is not trivial to write the input and read the output provided by one of such cellular automaton. The Density Classification Task (DCT) is a benchmark problem used to investigate which tasks may be performed by cellular automata in a parallel and distributed way. Evolutionary algorithms have been used to explore the cellular automata rule spaces in search for a possible partial solution for the problem. This project aims to use such an algorithm to search the update schedule space for improved versions of a cellular automaton rule found in the literature as a partial solution for the DCT.

Keywords: Cellular automata, genetic algorithms, density classification task.

1. Introdução

1.1 Problema de Pesquisa

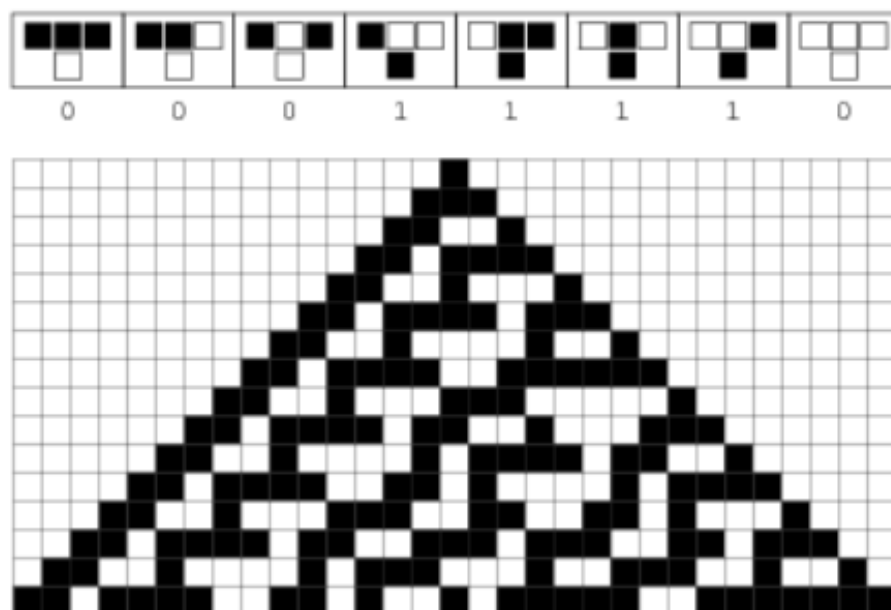
As bases para esse projeto de pesquisa foram fundamentadas por von Neumann e seu construtor universal, dispositivo detalhado em seu livro “Theory of Self Reproducing Automata” (VON NEUMANN; BURKS & GOLDSTINE, 1966). O objetivo de von Neumann era construir uma máquina que quando executada replicasse a si mesma. Segundo o *design* de von Neumann, a máquina conteria: um mecanismo que fosse capaz de ler e construir qualquer máquina especificada por um projeto, uma “máquina de cópias” capaz de copiar qualquer projeto e um projeto de si mesma. Assim sendo, existiriam duas maneiras para o dispositivo operar: (1) depois de construir a máquina especificada pelo projeto, a ferramenta de cópia seria usada para criar uma cópia do projeto, e esta cópia seria inserida na nova máquina, resultando numa versão completamente funcional da máquina anterior. (2) Ou o contrário, onde primeiro é feita a cópia do projeto, e depois construída a máquina.

A pesquisa de von Neumann dá origem ao campo dos autômatos celulares (ACs) durante o desenvolvimento da teoria do construtor universal e por sugestão de S. M. Ulman. O argumento de Ulman era o de que produzir as estruturas imaginadas por von Neumann seriam muito mais simples do ponto de vista lógico e matemático através de um *framework* celular.

O espaço no qual von Neuman planejava implementar seu autômato auto-reprodutor era muitas vezes referido pelo autor como um espaço de “cristalina regularidade”, “meio cristalino”, “estrutura granular” ou “estrutura celular” (VON NEUMANN; BURKS & GOLDSTINE, 1966). É neste espaço que foram desenvolvidas as pesquisas apresentadas aqui.

Autômatos celulares podem ser definidos como o agrupamento de muitos autômatos finitos conectados. De grande interesse para os pesquisadores são seus atributos espaço-temporais que por definição são discretos. Sua dinâmica é definida localmente por meio de regras de transição. Cada um dos autômatos finitos possui um conjunto de regras de transição que levam em conta também o estado dos autômatos vizinhos, dessa maneira, a computação ocorre apenas localmente, e nenhuma ação tomada de longa distância é possível. Portanto, o sistema possui leis locais e uniformes (onde “local” significa que o agente conhece apenas o que acontece onde ele está, num instante em que observa o estado de seus vizinhos mais próximos; E “uniforme” fala que as leis são as mesmas para todos os agentes). Na Figura 1 apresenta-se um autômato celular e sua função de transição, evoluído por 15 passos

Figura 1. Evolução de um autômato celular usando a regra de transição de número 30, a partir de um estado inicial com apenas a célula central no estado 1.



Desde as pesquisas de von Neumann, autômatos celulares têm sido aplicados em vários campos das ciências. Seus estudos podem ser aplicados na modelagem de sistemas reais como simulação de dinâmica de propagação de epidemias em populações (SIRAKOULIS, KARAFYLLIDIS & THANAILAKIS, 2000), sistemas físicos de dinâmica de fluidos (MARGOLUS, TOFFOLI & VICHNIAC, 1986), modelagem de tráfego de veículos (BOCCARA & FUK'S, 1999; NAGEL, WOLF, WAGNER & SIMON, 1997), entre outros.

Todavia, no que se refere ao esquema de atualização dos autômatos, a alternativa encontrada em um esquema assíncrono tem se mostrado de relevância para os diferentes modelos construídos, principalmente naqueles que pretendem reproduzir comportamentos de sistemas complexos reais, uma vez que eventos encontrados nesse tipo de modelo tendem a surgir da interação não homogênea entre os agentes do ponto de vista temporal.

Um problema de *benchmark* para testar a capacidade computacional dos autômatos celulares no que diz respeito a computação local é a chamada Tarefa de Classificação da Densidade TCD ou, em inglês, *Density Classification Task (DCT)*.

O problema de classificação da densidade consiste em decidir se a maioria das células da configuração binária (necessariamente de tamanho ímpar) estão no estado 0 ou 1 iterando-se um autômato celular sobre essa configuração. Uma regra que seja capaz de resolver tal problema deve convergir para uma configuração final que não mude mais, na qual todas as células estão no estado de maior presença na configuração inicial. (PACKARD, 1988).

1.2 Justificativa

Tendo-se em vista a importância da TCD como problema clássico e referencial para o estudo de tarefas globais que podem ser realizadas de maneira distribuída e com base apenas em informações locais, seu interesse do ponto de vista teórico-computacional é inegável.

É possível que técnicas utilizadas para a busca de soluções parciais ótimas possam ser generalizadas para outros problemas análogos para autômatos celulares. Segundo de Oliveira (2013), a tarefa de classificação de densidade é uma questão fundamental em outras áreas do conhecimento sob o ponto de vista de como fenômenos emergentes globais surgem a partir de ações locais.

Uma vez que a TCD não pode ser resolvida a partir de atualizações síncronas e sabendo-se que a assincronicidade é capaz de ampliar a riqueza de comportamentos dinâmicos para regras fixas, é natural considerar a possibilidade de solução perfeita ou, pelo menos, uma solução próxima a perfeita, para o problema em questão, considerando-se essa generalização na maneira como as células interagem.

Sendo impraticável fazer uma busca exaustiva por esquemas de atualização ótimos para configurações de tamanho arbitrário, a utilização de algoritmos evolutivos para realizar a busca de tais esquemas apresenta-se como um caminho viável, tendo sido anteriormente utilizados para busca de regras em contextos semelhantes (DE OLIVEIRA, BORTOT e OLIVEIRA, 2006; WOLZ e DE OLIVEIRA, 2008).

1.3 Objetivo

O objetivo deste trabalho foi buscar, por meio de um algoritmo evolutivo, por esquemas de atualização assíncrona para autômatos celulares que melhorassem o desempenho de soluções parciais da Tarefa de Classificação de Densidade obtidas para o caso síncrono.

Buscando o desenvolvimento do objetivo acima, estipulou-se os objetivos definidos abaixo:

- Realizar uma revisão bibliográfica buscando por trabalhos relacionados e técnicas evolutivas utilizadas para a busca de regras em espaços de autômatos celulares unidimensionais binários;
- Estabelecer um conjunto de regras que serão utilizadas para a busca por esquemas de atualização assíncrona;

- Avaliar os parâmetros utilizados para a busca evolutiva por esquemas de atualização assíncrona em espaços de configuração menores para os quais seja possível avaliar efetivamente cada esquema de atualização possível;
- Realizar buscas evolutivas em espaços de configurações maiores por esquemas de atualização que melhorem o desempenho de soluções parciais da tarefa de classificação de densidade para atualizações síncronas.

2. REFERENCIAL TEÓRICO

Desde os primeiros estudos de von Neumann (1966) envolvendo autômatos celulares no contexto de autorreplicação de estruturas, eles têm sido utilizados tanto para a simulação de fenômenos reais, como propagação de doenças (MIKLER, VENKATACHALAM e ABBAS, 2005), dinâmica dos fluidos (WOLF-GLADROW, 2004) e modelos de crescimento urbano (ABURAS, HO, *et al.*, 2016), quanto sob o ponto de vista de suas propriedades matemático-computacionais (MITCHELL, 1996; WOLFRAM, 1984). Wolfram (2002) conjectura que qualquer fenômeno emergente global pode ser explicado por e modelado a partir de estruturas simples que interagem localmente, o que inclui os ACs.

De fato, até mesmo em casos mais simples, como o dos autômatos celulares elementares (ACEs), ACs com conjunto de estados binário e comportamento localmente definido por funções que atuam sobre conjuntos de três *bits*, há regras, como a Regra 110 e suas equivalentes dinâmicas, capazes de simular uma máquina de Turing universal (COOK, 2004). Além delas, o AC conhecido como “*Conway’s Game of Life*” (GARDNER, 1970), uma regra binária bidimensional definida localmente com base em nove entradas binárias (a célula central de uma matriz e suas vizinhas mais próximas), também apresenta computabilidade universal.

Langton (1990) sugere que há regras com capacidade computacional de armazenamento e processamento de informações em uma região dos espaços de regras denominada beira do caos, localizada entre regras com comportamento ordenado e previsível e regras com comportamento aparentemente caótico.

Apesar de algumas regras apresentarem computabilidade universal, a codificação do problema e a leitura da saída produzida por regras como as citadas acima necessitam de acesso global ao conjunto de células, denominado configuração.

Sendo assim, uma questão natural que surge é: quais tarefas podem ser executadas por ACs considerando-se apenas o acesso e atividade locais sobre a configuração?

Diversos problemas desse tipo têm sido estudados pela comunidade científica ao longo dos anos, como a tarefa de sincronização (TS), o problema da paridade (PP) e a tarefa de classificação de densidade (TCD). A TS e o PP apresentam soluções perfeitas (MOORE e LANGDON, 1968; BETEL, DE OLIVEIRA e FLOCCHINI, 2013), enquanto que Land e Belew (1995) provaram que não existe uma solução desse tipo para a TCD.

Uma vez que é impossível obter uma solução perfeita para a definição clássica do problema, duas questões se apresentam naturalmente: qual a melhor solução possível para a definição padrão do problema? E quais modificações podem ser feitas nas regras de ACs para que uma solução perfeita ou arbitrariamente próxima à perfeita possam ser encontradas?

Quanto à primeira questão, diversas abordagens têm sido utilizadas para a obtenção de regras determinísticas capazes de produzir soluções progressivamente melhores para o problema em questão. Gacs, Kurdyumov e Levin (1978) foram os primeiros a propor uma regra, conhecida como GKL, capaz de resolver parcialmente um problema equivalente ao que viria a ser a TCD. Reynaga e Amthauer (2003) produziram uma generalização da GKL já no contexto da DCT. Buscas evolutivas por soluções mais próximas à perfeita foram realizadas em (OLIVEIRA e SIQUEIRA, 2006; MORALES, CRUTCHFIELD e MITCHELL, 2000; DE OLIVEIRA, BORTOT e OLIVEIRA, 2006; WOLZ e DE OLIVEIRA, 2008), com as melhores regras conhecidas até a presente data tendo sido obtidas por Wolz e de Oliveira (2008). Uma revisão abrangente desse tópico pode ser encontrada em (DE OLIVEIRA, 2013).

Já no contexto da segunda questão, Fatès (2011) mostrou que ACs estocásticos, isto é, aqueles para os quais a regra de interação local não é determinística, são capazes de resolver a TCD com precisão arbitrária. Além de regras estocásticas, foram também consideradas composições espaço-temporais de regras (SAHOO, CHOUDHURY, *et al.*, 2014), utilização de regras com memória (ALONSO-SANZ e BULL, 2009), por exemplo.

Faz sentido então considerar a possibilidade de uma solução perfeita ou melhor que as conhecidas atualmente ao considerar ACs com esquemas de atualização arbitrários, uma vez que do ponto de vista computacional, ACs assíncronos são capazes de apresentar maior diversidade de comportamento que os síncronos.

De fato, certas propriedades dinâmicas de alguns ACs são resultantes do esquema de atualização síncrona e não puramente da regra de interação local entre as células (VANNESCHI e MAURI, 2012; VIELHABER, 2013).

Uma vez que a quantidade de esquemas de atualização cresce exponencialmente com o tamanho das configurações consideradas (ARACENA, DEMONGEOT, *et al.*, 2013), é necessário utilizar-se algum método para a busca não-exaustiva de esquemas de atualização capazes de melhorar o desempenho de regras síncronas.

Os algoritmos evolutivos foram recorrentemente utilizados na TCD para buscas no espaço de regras. Mais precisamente, a proposta deste projeto é utilizar algoritmos genéticos para realizar buscas no espaço de esquemas de atualização.

Um algoritmo genético consiste em três processos básicos, iterados ao longo das gerações (DE CASTRO, 2007):

- Uma população formada por indivíduos com carga genética inicialmente randômica e capaz de se reproduzir de maneira que seus descendentes carreguem parte de sua carga genética;
- Variação genética devida a possíveis mutações que ocorram nos indivíduos de geração para geração;
- Seleção natural com base na adaptação de cada indivíduo ao ambiente, isto é, de maneira que a probabilidade de sobrevivência de certo indivíduo de uma geração para a próxima seja proporcional a seu nível de adaptação.

No contexto tratado, a carga genética de cada indivíduo é uma codificação de um esquema de atualização, a reprodução representa uma espécie de mistura entre esquemas de atualização distintos e a mutação ocasiona pequenas mudanças pontuais nas prioridades de atualizações das células de um certo esquema em questão. Por fim, o nível de adaptação será medido pelo desempenho que um esquema de atualização aplicado a uma regra previamente fixada obtém quando avaliado na TCD.

3. METODOLOGIA

Tanto o algoritmo evolutivo quanto o autômato celular foram implementados em linguagem C, devido ao custo computacional envolvido na obtenção da evolução temporal proveniente de regras de autômatos celulares assíncronos.

Para a implementação da estrutura computacional que representasse os esquemas de atualização assíncrona utilizou-se de um vetor simples composto por inteiros (vetor de atualização U). O tamanho desse vetor era o mesmo da configuração do AC e cada posição indicava uma prioridade de atualização. Essa prioridade era relativa à célula que estivesse na mesma posição no autômato. O vetor de atualização era preenchido com prioridades geradas de maneira aleatória, com o limite superior igual ao tamanho da configuração menos 1.

$$|C| = |U|$$

$$\max(U) \leq |C| - 1$$

Posições com a mesma prioridade são executadas ao mesmo tempo, e prioridades de menor valor são executadas antes das com maior valor.

Um exemplo de vetor de atualização para um autômato celular de tamanho 5, é:

$$U = (0, 1, 3, 0, 2)$$

No caso acima, a 1ª e a 4ª células seriam as primeiras a serem atualizadas (simultaneamente), seguidas da 2ª, da 5ª e da 3ª, sequencialmente.

A seguir, na Tabela 2 verifica-se um exemplo de aplicação desse vetor de atualização, executando-se um autômato celular com a regra de número 30 (cuja função de transição está descrita na Tabela 1). Células em vermelho representam uma regra recém aplicada, enquanto em azul, células cuja atualização já foi realizada em um passo anterior.

Tabela 1. Função de transição da regra 30.

Regra 30								
Célula atual e vizinhos	111	110	101	100	011	010	001	000
Novo estado para a célula central	0	0	0	1	1	1	1	0

Tabela 2. Demonstração da evolução de um AC assíncrono, dado o vetor de atualização U.

Passo	Estado	Regra(s) aplicada(s)
0	{1, 0, 1, 0, 1}	
1	{0, 0, 0, 0, 1}	(110) → 0, (101) → 0
2	{0, 0, 0, 0, 1}	(000) → 0
3	{0, 0, 0, 0, 1}	(010) → 1
4	{0, 0, 0, 0, 1}	(000) → 1

O passo 4 finaliza a atualização do autômato celular, uma vez que todas as células tiveram seu estado computado.

Alguns esquemas de atualização notáveis:

- Esquema de atualização síncrono; no caso de todas as prioridades serem iguais, o autômato celular se comporta de maneira equivalente a um AC síncrono.

- Esquema de atualização completamente assíncrono; se todas as células tiverem prioridade de atualização diferentes, o autômato terá uma dinâmica de atualização completamente assíncrona, sendo cada célula atualizada em um momento distinto.

O algoritmo evolutivo foi executado a partir da regra número 337607298446901146542393000444934784552 no espaço de regras binárias de raio 3 (WOLZ & DE OLIVEIRA, 2008), isto é, regras que recebem sete *bits* de entrada e retornam um *bit* de saída.

Cada indivíduo na população consistia em um vetor de atualização de comprimento 43, cada um de seus elementos tendo sido gerado aleatoriamente com distribuição uniforme no conjunto de estados possíveis (de 0 a 42). Cada indivíduo era então submetido a um possível processo de mutação: cada posição do vetor de atualização era percorrida e considerando-se uma probabilidade de mutação (variada de 1% a 10% entre os testes), sendo que, em caso de mutação, o valor naquela posição era alterado para outro inteiro aleatório dentro do intervalo possível (de 0 a 42).

Finalmente, para avaliar o *fitness* (aptidão) de cada indivíduo, iterou-se cada regra resultante sobre um conjunto de 100 configurações iniciais aleatórias de tamanho 43 e verificou-se em que proporção dessas configurações a regra convergiu para os pontos-fixos esperados. A seleção era feita então entre a população anterior e a população obtida após a mutação, mantendo-se fixo o tamanho da população para a próxima geração e fazendo a seleção pelo método da roleta, isto é, com chance de sobrevivência proporcional ao *fitness* do indivíduo. A cada nova geração o conjunto de configurações iniciais era mantido o mesmo.

Variou-se o tamanho da população entre 100, 200 e 300 vetores de atualização e a taxa de mutação de 1% a 10%, com intervalos de 1%.

4. RESULTADO E DISCUSSÃO

Os resultados dos passos descritos anteriormente podem ser observados na Tabela 3, a seguir.

Tabela 3. Comparação de desempenho encontrado a partir da variação dos parâmetros de mutação intermediária e tamanho da população.

Mutação (%)	Tamanho da população	Melhor desempenho (%)	Média melhores desempenhos (%)
1	100	91	86.12

	200	95	86.75
	300	91	86.81
2	100	92	84.93
	200	91	84.57
	300	96	85.89
3	100	93	83.99
	200	92	86.95
	300	93	88.26
4	100	90	81.74
	200	90	82.7
	300	92	84.79
5	100	89	84.91
	200	90	82.42
	300	87	81.29
6	100	95	86.35
	200	90	85.52
	300	93	86.1
7	100	92	80.51
	200	87	79.83
	300	89	83.12
8	100	87	82.61
	200	90	82.55
	300	92	87.13
9	100	87	82.16
	200	89	82.98
	300	92	89.01
10	100	91	84.56
	200	92	86.13
	300	91	85.65

Aplicando-se os testes de correlação de Pearson e de Spearman (Tabelas 4 e 5), os resultados não foram satisfatórios para atribuir-se correlação linear forte entre a mutação e o maior desempenho da evolução ou até mesmo com o desempenho médio dos indivíduos de maior *fitness* a cada geração.

O mesmo pode ser dito ao tentar estipular uma correlação a partir do tamanho da população no que tange o melhor desempenho encontrado, porém é possível admitir uma correlação fraca com a média dos melhores desempenhos.

Isso possivelmente deve-se ao fato da população ser relativamente pequena para o espaço de busca (a quantidade de esquemas de atualização independentes cresce exponencialmente de acordo com o tamanho da configuração), bem como a quantidade de configurações iniciais para teste ser também bastante pequena (100 num espaço de 2^{43} possíveis).

Tabela 4. Teste de correlação de Pearson.

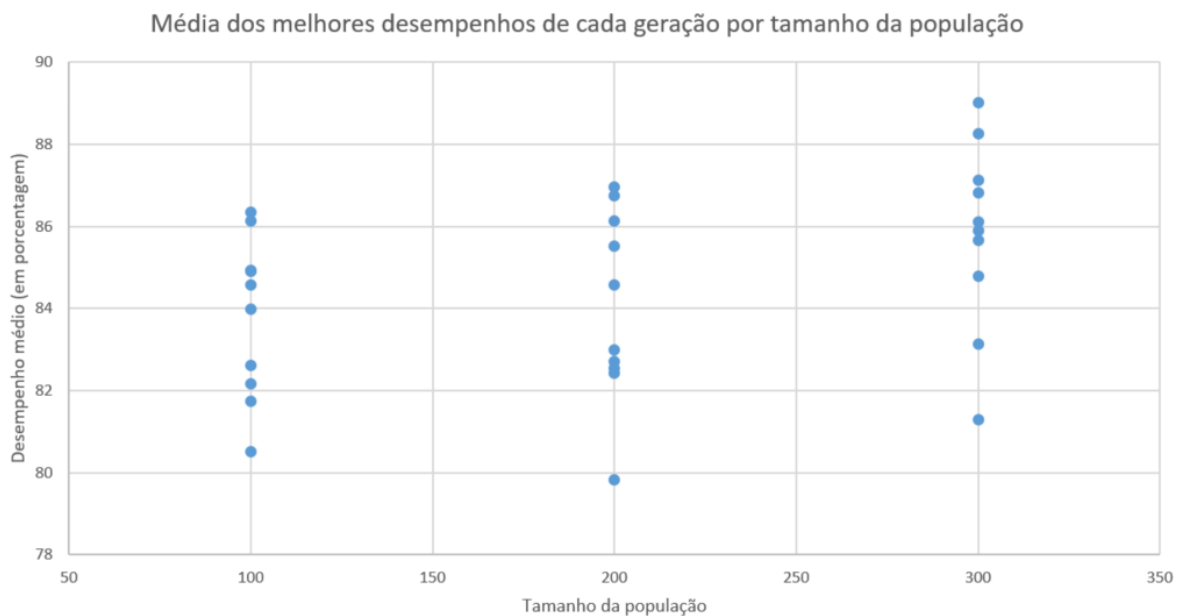
Teste de correlação de Pearson		
	Melhor desempenho de toda a evolução	Média dos melhores desempenhos de cada geração
Taxa de mutação	-0.377976634	-0.207896061
Tamanho da população	0.16062894	0.363104298

Tabela 5. Teste de correlação de Spearman.

Teste de Correlação de Spearman		
	Melhor desempenho de toda a evolução	Média dos melhores desempenhos de cada geração
Taxa de mutação	-0.345683084	-0.223242453
Tamanho da população	0.172014654	0.372616638

O gráfico a seguir ajuda a ilustrar a correlação fraca encontrada entre o tamanho da população e a média dos melhores desempenhos de cada geração.

Figura 2. Média dos melhores desempenhos de cada geração por tamanho da população



5. CONSIDERAÇÕES FINAIS

Tendo-se em vista os resultados acima, nota-se como a busca por meio do algoritmo genético permitiu encontrar esquemas de atualização com desempenhos melhores que os normalmente encontrados para o caso síncrono da regra 337607298446901146542393000444934784552, que ficam normalmente em torno de 80%.

Entretanto, deve-se levar em conta que as amostras de configurações iniciais utilizadas são bastante pequenas em relação ao tamanho de todo o espaço possível. Uma vez que o processo de obtenção de evoluções de autômatos celulares assíncronos é custoso, seria necessário utilizar sistemas computacionais de maior potência para verificar de maneira conclusiva a robustez dos resultados.

Mesmo assim, a busca evolutiva mostrou-se como uma ferramenta viável para explorar o espaço de esquemas de atualização em busca de otimizar uma regra síncrona que já apresenta bom desempenho na tarefa de classificação de densidade.

Como trabalhos futuros, sugere-se a aplicação do processo descrito neste artigo utilizando-se populações e conjuntos de avaliação maiores, além da possibilidade de explorar outros espaços de assincronia que não o determinístico aqui descrito, como o de assincronia condicional ou de assincronia probabilística.

6. AGRADECIMENTOS

Gostaria de agradecer ao apoio oferecido pelo Mackpesquisa através da bolsa de iniciação científica, tendo ela sido de suma importância para o desenvolvimento deste projeto.

Agradeço também ao meu orientador, Prof. Eurico Ruivo, por toda a paciência e dedicação.

7. REFERÊNCIAS

ABURAS, M. et al. The simulation and prediction of spatio-temporal urban growth trends using cellular automata models: A review. **International Journal of Applied Earth Observation and Geoinformation**, 52, 2016. 380-389.

ALONSO-SANZ, R.; BULL, L. A very effective density classifier two-dimensional cellular automaton with memory. **Journal of Physics A**, 42, n. 48, 2009. 485101.

ARACENA, J. et al. On the number of update digraphs and its relation with the feedback arc sets and tournaments. **Discrete Applied Mathematics**, 161, n. 10, 2013. 1345-1355.

BETEL, H.; DE OLIVEIRA, P.; FLOCCHINI, P. Solving the parity problem in one-dimensional cellular automata. **Natural Computing**, 12, n. 3, 2013. 323-337.

BOCCARA, N., & FUK'S, H. (1999). **Number-conserving Cellular Automaton Rules. Fundamenta Informaticae**, 52, 1–13.

COOK, M. Universality in elementary cellular automata. **Complex Systems**, 15, n. 1, 2004. 1-40.

DE CASTRO, L. Fundamentals of natural computing: an overview. **Physics of Life Reviews**, 4, 2007. 1-36.

DE OLIVEIRA, P. Conceptual connections around density determination in cellular automata. In: KARI, J.; KUTRIB, M.; MALCHER, A. **International Workshop on Cellular Automata and Discrete Complex Systems**. Berlin: Springer, 2013. p. 1-14.

DE OLIVEIRA, P.; BORTOT, J.; OLIVEIRA, G. The best currently known class of dynamically equivalente cellular automata rules for density classification. **Neurocomputing**, 70, n. 1-3, 2006. 35-43.

FATÈS, N. **Stochastic cellular automata solve the density classification problem with an arbitrary precision**. Symposium on Theoretical Aspects of Computer Science - STACS 2011. Dortmund: Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik. 2011. p. 284-295.

GACS, P.; KURDYUMOV, G.; LEVIN, L. One-dimensional uniform arrays that wash out finite islands. **Problemy Peredachi Informatsii**, 14, 1978. 92-98.

GARDNER, M. Mathematical games - The fantastic combinations of John Conway's new solitaire game, "Life". **Scientific American**, v. 223, p. 120-123, 1970.

LAND, M.; BELEW, R. No perfect two-state CA for density classification task exists. **Physical Review Letters**, 74, n. 25, 1995. 5148-5150.

LANGTON, C. Computation at the edge of chaos: Phase transitions and emergent computation. **Physica D: Nonlinear phenomena**, 42, n. 1-3, 1990. 12-37.

MARGOLUS, N., TOFFOLI, T., & VICHNIAC, G. (1986). **Cellular-automata Supercomputers for Fluid-dynamics Modeling**. Physical Review Letters, v56(n16), p1694.

MIKLER, A.; VENKATACHALAM, S.; ABBAS, K. Modeling infectious diseases using global stochastic cellular automata. **Journal of Biological Systems**, 13, n. 4, 2005. 421-439.

MITCHELL, M. Computation in cellular automata: a selected review. **Nonstandard computation**, 1996. 95-149.

MOORE, F.; LANGDON, G. A generalized firing squad problem. **Information and control**, 12, 1968. 212-220.

MORALES, F.; CRUTCHFIELD, J.; MITCHELL, M. Evolving two-dimensional cellular automata to perform density classification: a report on work in progress. **Parallel Computing**, 27, 2000. 571-585.

OLIVEIRA, G.; SIQUEIRA, S. Parameter characterization of two-dimensional cellular automata rule space. **Physica D**, 217, n. 1, 2006. 1-6.

PACKARD, N. Adaptation towards the edge of the chaos. In: KELSO, J.; MANDELL, A.; SHLESINGER, M. **Dynamic Patterns in Complex Systems**. Singapore: World Scientific, 1988. p. 293-301.

REYNAGA, R.; AMTHAUER, E. Two-dimensional cellular automata of radius one for density classification task $p=1/2$. **Pattern Recognition Letters**, 24, n. 15, 2003. 2849-2856.

SAHOO, S. et al. Solutions on 1-D and 2-D Density Classification Problem Using Programmable Cellular Automata. **Journal of Cellular Automata**, 9, n. 1, 2014. 59-88.

SIRAKOULIS, G., KARAFYLLIDIS, I., & THANAILAKIS, A. (2000). **A Cellular Automaton Model for the Effects of Population Movement and Vaccination on Epidemic Propagation**. Ecological Modelling, 133(3), 209–223.

VANNESCHI, L.; MAURI, G. A study on learning robustness using asynchronous 1D cellular automata rules. **Natural Computing**, 12, n. 3, 2012. 289-302.

VIELHABER, M. Computation of functions on n bits by asynchronous clocking of cellular automata. **Natural Computing**, 12, n. 3, 2013. 307-322.

VON NEUMANN, JOHN, ARTHUR W. BURKS, AND HERMAN H. GOLDSTINE. (1966). **Theory of self-reproducing automata**. Urbana: University of Illinois Press.

WOLF-GLADROW, D. **Lattice-gas cellular automata and lattice Boltzmann models: an introduction**. Berlin: Springer, 2004.

WOLFRAM, S. Computation theory of cellular automata. **Communications in mathematical physics**, 96, n. 1, 1984. 15-57.

WOLFRAM, S. **A new kind of science**. Champaign - IL: Wolfram Media, 2002.

WOLZ, D.; DE OLIVEIRA, P. Very effective evolutionary techniques for searching cellular automata rule spaces. **Journal of Cellular Automata**, 3, n. 4, 2008. 289-312.

Contatos: osnielteixeira@icloud.com e eurico.ruivo@mackenzie.br