

# Previsão dinâmica e recurso compartilhado auxiliando a busca evolutiva de autômatos celulares bidimensionais

Gina M. B. de Oliveira e Sandra R. C. Siqueira

**Abstract**— Autômatos celulares (ACs) são sistemas dinâmicos discretos que exibem uma diversidade de comportamentos dinâmicos. Uma característica dos ACs é a sua capacidade de executar computações através de processamentos locais e intrinsecamente paralelos. Entretanto, uma vez especificada a tarefa que o AC deve resolver, não é trivial encontrar a regra de transição adequada, devido à alta cardinalidade do espaço de busca. Uma solução tem sido a aplicação de algoritmos genéticos (AGs) na busca das regras. Em trabalhos anteriores o AG foi auxiliado por uma heurística baseada em parâmetros de previsão de comportamento dinâmico dos ACs unidimensionais. Nesse trabalho, três parâmetros de previsão foram generalizados para os ACs bidimensionais. Eles foram utilizados na versão bidimensional da tarefa da classificação da densidade. Além disso, a estratégia recurso compartilhado foi adicionada ao AG. Assim, foi possível encontrar uma regra com eficácia superior à melhor regra publicada para essa tarefa.

**Index Terms**— Autômato celular, algoritmo genético, parâmetros de previsão de dinâmica, tarefa da classificação da densidade bidimensional.

## I. INTRODUÇÃO

AUTÔMATOS Celulares (ACs) são sistemas dinâmicos totalmente discretos: tempo, espaço e variáveis discretas.

Apesar de serem formados por componentes extremamente simples os ACs exibem uma variedade dinâmica [18], desde o comportamento ponto-fixo homogêneo ao caótico. A dinâmica de um AC está associada à sua regra de transição. Pesquisadores têm se empenhado em estudar indicadores, calculados diretamente sobre as regras dos ACs, que aproximam a previsão do seu comportamento dinâmico [1,7,13]. A maioria dos parâmetros referem-se aos ACs unidimensionais.

Uma característica que tem chamado a atenção dos pesquisadores é a capacidade dos ACs executarem computações através de um comportamento emergente. Métodos evolutivos têm sido utilizados na busca das regras de transição que executem uma tarefa computacional [5,9,11,12,16]. Uma das mais estudadas é a tarefa de classificação da densidade (TCD) [9], na qual o AC deve decidir se o reticulado possui mais

células no estado 0. A maioria dos trabalhos se refere à versão unidimensional da TCD.

Em trabalhos anteriores, heurísticas baseadas em parâmetros de previsão dinâmica foram agregadas à busca evolutiva na TCD unidimensional, aumentando a eficiência do método [12]. Nesse trabalho, três parâmetros de previsão foram generalizados do contexto unidimensional para o bidimensional. São eles: sensibilidade, domínio da vizinhança e propagação da atividade. Posteriormente, eles foram utilizados para auxiliar a busca por regras que resolvessem a versão bidimensional da TCD. Os parâmetros foram incorporados em um AG com a técnica recurso compartilhado (RC) [3]. Assim, foi possível encontrar uma regra com eficácia superior à melhor regra publicada para a TCD bidimensional [11].

O restante do artigo está dividido em cinco seções. A próxima seção discute a sistemática dos AGs e a técnica de nicho utilizada. A seção 3 apresenta aspectos básicos dos ACs e seu comportamento dinâmico. Na seção 4, a utilização de métodos evolutivos na busca de regras que resolvam a TCD é discutida. A generalização bidimensional dos três parâmetros é apresentada na seção 5. Os resultados dos experimentos envolvendo a heurística dos parâmetros bidimensionais e a técnica de recurso compartilhado são discutidos na seção 6. A última seção apresenta algumas considerações finais sobre o trabalho.

## II. AG E RECURSO COMPARTILHADO

O método de busca conhecido como Algoritmo Genético (AG) proposto Holland na década de 60 é uma metáfora computacional do processo de evolução biológica [4]. No AG, parte-se de uma população de indivíduos (soluções candidatas), que são avaliadas de forma a se medir o quão próximas elas estão da solução do problema [3]. Baseado nesta avaliação, ou função de aptidão, um processo de seleção separa um subconjunto desta população para servir de base para a geração de novas soluções candidatas, obtidas através da aplicação dos operadores *crossover* e *mutação* sobre o subconjunto selecionado. Este processo se repete por várias iterações, ou gerações, e a população evolui até que se chegue em soluções satisfatórias.

Os AGs têm sido empregados por sua capacidade de encontrar uma solução adequada, considerando uma pequena amostra do espaço de busca. Mas, existem casos em que o esse espaço é muito extenso, dificultando a convergência. Algumas técnicas

Este trabalho foi parcialmente auxiliado pelo CNPq (PQ 304639/2004-4), CAPES e MACKPESQUISA.

G. M. B. Oliveira é professora da Faculdade de Computação (FACOM), Universidade Federal de Uberlândia, Av. João Naves D'Ávila 2160, Bloco B, Uberlândia, MG, 38400-902, email: gina@facom.ufu.br.

S. R. C. Siqueira é professora da Faculdade de Computação e Informática (FCI), Universidade Presbiteriana Mackenzie, Rua da Consolação 896, São Paulo, SP, 01302-907, email: sandra@mackenzie.com.br.

conhecidas por estratégias de avaliação competitiva para a preservação da diversidade [17], ou simplesmente, técnicas de nicho, foram elaboradas para melhorar a convergência dos AGs. Existem várias técnicas de nicho [5]. Entre elas, o recurso compartilhado (RC) no qual são valorizados os indivíduos que solucionam instâncias do problema não resolvidas por outros.

Para exemplificar, suponha que a Tabela 1 apresenta uma população formada por 5 soluções ( $S_1 S_2 S_3 S_4 S_5$ ) e que a avaliação é realizada através do confronto dessas soluções com 5 instâncias do problema a ser resolvido ( $I_1 I_2 I_3 I_4 I_5$ ). O sucesso no confronto da solução  $S_i$  com a instância  $I_j$  é denotado por 1 e o fracasso por 0. Em um AG simples, a avaliação de cada solução é dada pela soma dos confrontos, correspondente à linha  $S^*$  da tabela. As soluções  $S_3$  e  $S_4$  têm a mesma avaliação, igual a 4. No RC, leva-se em consideração não apenas o número de instâncias resolvidas pela solução, mas também o nível de dificuldade de cada uma. A coluna  $I^*$  apresenta o número de fracassos das soluções associados a cada instância. Assim, vemos que a instância  $I_2$  é a mais difícil de ser resolvida, uma vez que a mesma só foi resolvida pela solução  $S_3$ . Assim, os valores da coluna  $I^*$  são utilizados como pesos na contabilização da avaliação das soluções. Dessa forma, no RC, a avaliação da solução  $S_3$  é 7 e a da solução  $S_4$  é 5, refletindo o fato de  $S_3$  resolver instâncias do problema mais difíceis. Com o uso do RC na avaliação, existe uma tendência de que os casos mais difíceis sejam resolvidos, evitando que soluções irrelevantes para o problema dominem a população.

TABELA I  
CONFRONTOS  $S_i \times I_j$

	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	$I^*$
$I_1$	1	0	1	1	1	1
$I_2$	0	0	1	0	0	4
$I_3$	1	1	1	1	1	0
$I_4$	1	1	0	1	0	2
$I_5$	0	1	1	1	0	2
$S^*$	3	3	4	4	2	

### III. AUTÔMATOS CELULARES

Um AC é composto pelo espaço celular e pela regra de transição. O espaço celular é representado por um reticulado formado por  $n$  células, cada uma em um estado entre  $k$  possíveis. Os mais estudados são os unidimensionais (AC 1D), que consideram reticulados cujas células estão posicionadas linearmente, e os bidimensionais (AC 2D), que consideram reticulados cujas células são organizadas sob a forma de uma grade. A evolução espaço-temporal de um AC consiste em modificar o estado de cada célula ao longo do tempo, de acordo com os estados correntes e com a regra de transição. A regra de transição estabelece o novo estado da célula, dado o estado atual da célula e de suas vizinhas. A Figura 1 apresenta a evolução por um passo de tempo do reticulado de um AC 1D binário com a vizinhança formada por 3 células. A regra de transição é formada

por 8 bits, que indicam qual deverá ser o novo estado da célula central para cada uma das  $2^3$  vizinhanças possíveis. A atualização da oitava célula é destacada: os estados das células da vizinhança são 0 0 1 e a regra estabelece que o novo valor será 1. O mesmo processo é realizado para todas as células do reticulado. A seqüência dos reticulados ao longo do tempo define o diagrama espaço-temporal. No caso 1D, o diagrama resultante é um plano. No caso 2D, é uma seqüência de planos. As figuras 2 e 3 apresentam diagramas espaço-temporais de ACs 1D. A Figura 4 apresenta alguns passos da evolução temporal de um AC 2D. O estado 0 é representado em branco e o estado 1 em preto. No caso 1D, utiliza-se um parâmetro chamado raio que indica a quantidade de células consideradas na vizinhança em relação à central. Nesse trabalho, foram utilizados os AC 2D binários, definidos pela vizinhança de Moore, que é formada por nove células [14], dada pela célula central e as oito vizinhas que a tocam no plano.

Uma característica importante dos ACs é a diversidade de comportamentos dinâmicos [18]. Li e Packard propuseram um esquema de classificação que divide o espaço de regras em nulas, ponto-fixo, ciclo-duplo, periódicas, caóticas e complexas [8]. A Figura 2 apresenta uma evolução espaço-temporal de um AC 1D para cada classe Li-Packard. A dinâmica de um AC está associada à sua regra de transição. Pesquisadores têm se empenhado em estudar a dinâmica dos ACs através de indicadores, calculados sobre as regras, que aproximam a previsão do seu comportamento [1,13]. A maioria dos parâmetros focam-se nos ACs 1D. Três parâmetros de previsão generalizados para o contexto bidimensional, são apresentados na seção 5.

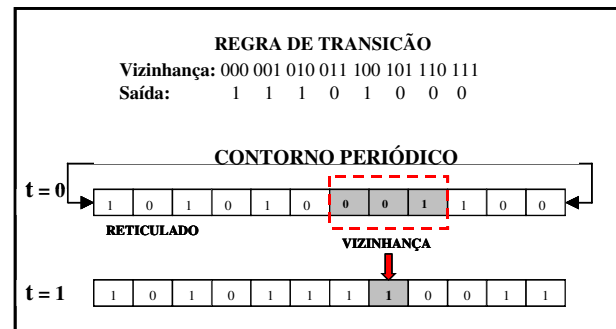


Fig. 1. Regra de transição e reticulado ( $t=0$  e  $t=1$ ) para um AC unidimensional binário.

### IV. EVOLUÇÃO DE AUTÔMATOS CELULARES NA TCD

Um aspecto bastante estudado diz respeito a como os ACs realizam computações, através de processamentos locais e intrinsecamente paralelos que interagem entre si, emergindo um comportamento global e coordenado. Os pesquisadores estão interessados nas relações entre o comportamento dinâmico genérico de um AC e suas habilidades computacionais, como parte de uma questão maior sobre as relações entre a teoria dos sistemas dinâmicos e a teoria computacional [18]. Os estudos sobre o poder computacional dos ACs podem ser divididos em duas abordagens distintas: a computação explícita e a computação implícita [10].

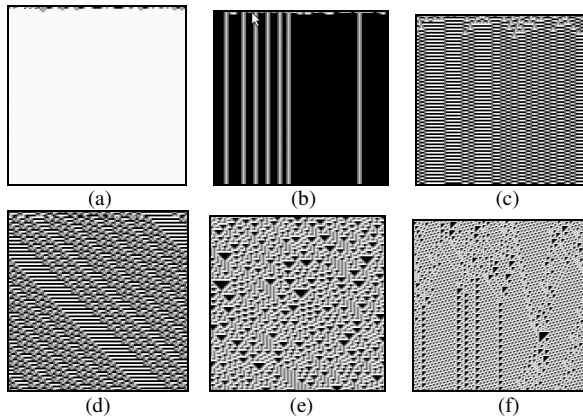


Fig. 2. Classes de comportamento dinâmico de ACs: (a) nulo (b) ponto-fixo (c) ciclo-duplo (d) periódico (e) caótico (f) complexo.

A computação implícita é aquela executada diretamente da regra do AC, onde a configuração inicial do reticulado é a entrada da computação e a saída é dada pela configuração do reticulado após um certo tempo. Já a computação explícita ocorre em um nível mais alto; o papel da regra do AC é prover a dinâmica de sustentação sobre a qual a computação será executada [10].

O entendimento de como os ACs executam a computação implícita ainda é vago. A maioria das pesquisas nesta abordagem estuda como os ACs são capazes de executar tarefas computacionais pré-especificadas. Para resolver tais tarefas, as regras devem ser capazes de propagar informações por grandes distâncias, uma vez que as vizinhanças são locais e a entrada da tarefa está distribuída por todo o reticulado. O principal exemplo é a resolução da tarefa de classificação da densidade (TCD) [9]. Neste problema, a meta é encontrar um AC 1D binário que decida se uma configuração inicial de  $N$  células com estados binários, possui uma quantidade maior de 1's. Se for maior, o reticulado deste AC deve evoluir para uma configuração ponto-fixo de 1's (as  $N$  células permanecerão no estado 1); caso contrário, uma configuração ponto-fixo de 0's. A Figura 3 apresenta duas evoluções de uma regra de transição de um AC unidimensional com vizinhança  $r=3$  e reticulado  $N=149$  células, executando perfeitamente a TCD. Na primeira evolução, a configuração inicial possui uma quantidade de 1's menor do que a de 0's. Na segunda evolução, a configuração inicial possui uma quantidade de 1's maior.

Projetar um algoritmo que execute a TCD é trivial para um sistema com um controlador central ou com armazenagem central, como em um computador com um contador ou em uma rede neural, onde todas as entradas são conectadas a uma unidade central da camada escondida. No entanto, esta é uma tarefa não trivial para ACs, uma vez que estes se baseiam apenas em interações locais para realizar sua evolução. Para executar bem esta tarefa, é requerido mais poder computacional do que o de uma simples célula ou uma combinação linear de células. Como as células 1's podem estar distribuídas ao longo do reticulado, a regra deve ser capaz de transferir informações por grandes distâncias. Para isto, é necessária uma coordenação

global das células que estão distantes e não podem se comunicar diretamente. Para a realização da TCD, é necessário explorar o paralelismo intrínseco dos ACs. Foi provado que nenhum AC de estado e raio finitos pode executar a TCD para todos os tamanhos de reticulado [6]. Entretanto, permanece em aberto qual o limite superior de eficácia que uma regra pode atingir.

Uma vez definida uma tarefa computacional, não é trivial encontrar um AC que a execute. A programação de um AC através de uma busca exaustiva no espaço de regras se torna impossível, devido à sua alta cardinalidade. A abordagem de maior êxito é a utilização de métodos evolutivos na busca de regras que computem uma tarefa especificada.

Packard foi o primeiro pesquisador a publicar resultados utilizando AGs como ferramenta para encontrar regras de ACs que resolvessem uma tarefa computacional, no caso, a TCD [16]. Ele considerou as regras como indivíduos de um AG. Para um AC binário com vizinhança de raio  $r$ , o número de *bits* de uma regra de transição é dado por  $2^{2r+1}$  e o espaço de regras é dado por  $2^{(2^{2r+1})}$ . Tipicamente, uma população representa somente uma pequena amostra deste espaço. A aptidão é dada pela habilidade do AC em executar a tarefa computacional. O *crossover* é dado pela geração de duas novas regras (filhos), a partir de segmentos de outras regras da população (pais). A mutação é dada pela alteração aleatória do *bit* de saída de uma das transições da regra. Vários estudos têm sido publicados onde a programação de um AC é conseguida através de uma busca evolutiva [5,11,12,17]. A evolução de ACs na busca da solução da TCD é o principal problema estudado. Os experimentos são realizados normalmente no espaço dos ACs 1D, binários e de raio 3. Em [11], um AG foi utilizado com o objetivo de encontrar regras que resolvessem uma generalização da TCD para o caso bidimensional, que chamaremos de TCD 2D. A Figura 4 apresenta um exemplo de evolução temporal de um AC resolvendo a TCD 2D. O reticulado inicial possui mais células no estado 0. No tempo 545, o AC converge todas as células do reticulado para o estado 0.

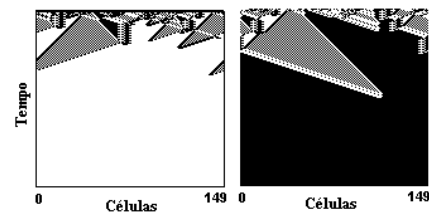


Fig. 3. Exemplos de execução da tarefa de classificação da densidade (AC 1D de raio 3).

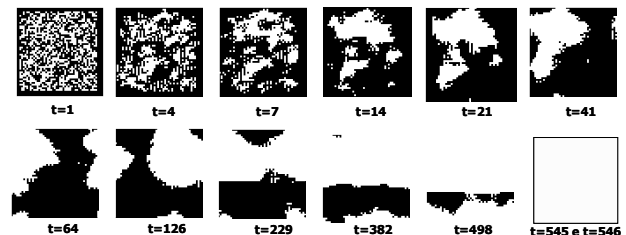


Fig. 4. Exemplo de execução da TCD 2D (vizinhança de Moore).

## V. GENERALIZAÇÃO DOS PARÂMETROS DE PREVISÃO PARA O CONTEXTO BIDIMENSIONAL

Em [12], um conjunto de parâmetros de previsão foi selecionado e aplicado na TCD 1D, auxiliando a busca evolutiva realizada pelo AG. Foi possível comprovar que os parâmetros mais importantes na melhoria da busca foram a sensibilidade, o domínio da vizinhança e a propagação da atividade [13]. Por isso, esses parâmetros foram escolhidos para a generalização 2D. Uma discussão detalhada sobre os parâmetros e suas definições formais são apresentadas em [15]. A idéia de todos os parâmetros é realizar um cálculo simples analisando-se todas as transições das vizinhanças que compõem uma regra. Os conceitos subjacentes aos três parâmetros no espaço unidimensional [13] foram mantidos na generalização proposta. No caso dos ACs 2D com vizinhança de Moore (9 células), uma regra é formada por 512 vizinhanças diferentes, representadas linearmente de 000-000-000 a 111-111-111.

A sensibilidade ( $s$ ) é calculada complementando-se cada célula da vizinhança em consideração e analisando-se o comportamento da saída de cada vizinhança complementar. Por exemplo, para a vizinhança 000-000-000, serão analisadas 9 vizinhanças complementares: 000-000-001 a 100-000-000. Suponha a vizinhança 000-000-000 com saída 1. Nesse caso, todas as vizinhanças complementares que possuem saída 0 serão considerados “sensíveis”. O parâmetro calcula o número total de vizinhanças complementares que são sensíveis. O valor máximo de contabilizações desse parâmetro no caso 2D com vizinhança de Moore é 4608 e este valor é utilizado para normalizar o parâmetro de 0 a 1.

O domínio da vizinhança ( $dm$ ) verifica se a saída de uma determinada vizinhança acompanha as células que a dominam. Por exemplo, para a vizinhança 010-001-110, o estado que domina a vizinhança é o estado 0. Se a saída dessa vizinhança for 1, não houve o domínio da vizinhança; ao contrário, se for 0, houve o domínio. Em seguida, é atribuído um peso a cada vizinhança analisada, inversamente proporcional à frequência com que um tipo de vizinhança ocorre na regra. Por exemplo, a vizinhança totalmente homogênea 000-000-000 tem o maior peso, 126, enquanto vizinhanças mais heterogêneas (por exemplo, 010-110-001 e 100-010-111) possuem o menor peso, igual a 1. O parâmetro contabiliza todas as ocorrências de domínio da vizinhança, analisando-se cada vizinhança individualmente e aplicando-se o peso correspondente. O valor máximo de contabilizações, ponderadas pelos respectivos pesos, é 6120. Esse valor também é utilizado na normalização entre 0 a 1.

A propagação da atividade ( $pa$ ) mistura os conceitos de sensibilidade e domínio da vizinhança. A princípio, verifica-se quais vizinhanças não sofrem domínio da vizinhança, que são consideradas ativas. Então, contabiliza-se para cada vizinhança classificada como ativa, quantas das 9 complementares permanecem ativas. Suponha uma vizinhança 000-000-000 cuja saída é 1: como não houve domínio da vizinhança, ela é considerada ativa. A seguir, a vizinhança complementar 000-000-001 é analisada. Se sua saída também for 1, essa vizinhança também é ativa e contabiliza-se 1 no cálculo do parâmetro. As outras 8 vizinhanças complementares são analisadas e, para cada

caso de atividade, contabiliza-se mais 1. Essa mesma análise é feita para todas as 512 vizinhanças da regra e suas 9 complementares. A cada ocorrência de atividade na vizinhança analisada e na sua complementar é somado 1 no cálculo do parâmetro. Se uma das duas não for ativa, o par não é contabilizado. O valor máximo de contabilizações desse parâmetro é 4608, utilizado na normalização do parâmetro.

## VI. EXPERIMENTOS

Inicialmente, o experimento descrito em [11] foi reproduzido. Os indivíduos são regras de transição de ACs 2D com vizinhança de Moore. A vizinhança é formada por 9 células e cada indivíduo é uma seqüência de 512 bits. A avaliação é dada pela eficácia da regra em classificar corretamente uma amostra de 100 reticulados: grades binárias formadas por  $21 \times 21$  células. A cada geração, uma amostra diferente é sorteada. A avaliação do indivíduo corresponde ao número de sucessos da regra na classificação correta da densidade dos 100 reticulados. Portanto, a avaliação é um inteiro entre 0 e 100. Uma taxa de elitismo de 20% é empregada de uma geração para outra. Os 80% restantes da população são formados pelos filhos gerados pelo *crossover* (ponto simples) de pares de pais selecionados aleatoriamente da elite. Uma taxa de mutação de 2% é aplicada a cada filho. O resultado final da evolução do AG é dado pela melhor regra obtida na última geração (após 100 gerações).

Os resultados obtidos foram compatíveis com o trabalho original. A eficácia da regra obtida ao final da execução do AG é medida pelo desempenho da regra na classificação de uma amostra de 10.000 reticulados iniciais. A média de eficácia na resolução da TCD 2D das regras encontradas em 100 execuções do AG foi de 56,58% e a melhor regra encontrada teve 67,57% de eficácia. A análise da regra encontrada a cada execução mostrou que foram caracterizados dois grandes grupos de regras: o primeiro formado por 22 regras de ciclo duplo, nas faixas de eficácia entre 43% e 55%, e o segundo formado por 40 regras nulas com eficácia entre 60% e 68%. O comportamento nulo é caracterizado por regras que convergem para uma situação homogênea: ou todas as células vão para o estado 0 ou todas vão para o estado 1. Em seguida, os três parâmetros de previsão de comportamento generalizados (sensibilidade, propagação da atividade e domínio da vizinhança) foram calculados para cada uma das 100 regras obtidas. A Tabela 2 apresenta as faixas encontradas para os parâmetros das regras em relação aos dois grupos: regras nulas e ciclo-2. Comparando-se os valores, pode ser observado que, para a sensibilidade, as regras nulas estão concentradas numa faixa inserida na obtida para as regras ciclo-duplo; para o domínio da vizinhança e a propagação da atividade, os grupos estão em faixas distintas. Além disso, as regras nulas foram caracterizadas em faixas bem específicas dos domínios dos parâmetros, correspondendo aproximadamente a apenas 5% do domínio de cada parâmetro, que varia de 0 a 1. Como o comportamento desejado na TCD 2D é o nulo, as faixas obtidas para as regras nulas foram consideradas como sendo as faixas desejáveis para uma regra mais eficaz. A informação da faixa desejável foi traduzida em uma função de forma semelhante ao que já havia sido feito no caso 1D [13]. A função retorna o valor máximo quando os parâmetros de uma regra

estão dentro das faixas desejadas e vai decrescendo à medida que os parâmetros se distanciam dessas faixas.

A heurística dos parâmetros foi incorporada no AG: a função de avaliação passou a utilizar uma parcela referente à heurística e os operadores mutação e *crossover* fazem algumas tentativas e selecionam as regras com maior valor segundo a heurística. Foram feitas 7 simulações utilizando diferentes combinações: individualmente, dois a dois e os três parâmetros simultaneamente. Cada simulação foi formada por 100 execuções do AG. Duas simulações retornaram os melhores resultados. A primeira utilizou a sensibilidade e a propagação da atividade com uma eficácia média de 60,95%. A segunda utilizou o domínio da vizinhança e a propagação da atividade com uma eficácia média de 65,03 %. Ambas com uma eficácia média superior à da busca sem os parâmetros (56,58%).

Posteriormente, um AG foi implementado incorporando a técnica recurso compartilhado (RC), que já havia se mostrado eficiente no caso da TCD 1D [17]. Na Tabela 3, são apresentados os resultados de 3 simulações. A primeira não utilizou qualquer informação dos parâmetros -  $AGRC(0)$ . As demais foram realizadas utilizando-se as 2 combinações que retornaram os melhores resultados no AG simples: sensibilidade e propagação da atividade -  $AGRC(s+pa)$  e domínio da vizinhança e propagação da atividade -  $AGRC(dm+pa)$ . Cada simulação foi formada por 100 execuções e ao final de cada execução a melhor regra da última geração foi extraída e sua eficácia foi avaliada na classificação de uma amostra de 10.000 reticulados iniciais. As 300 regras obtidas possuem eficácia entre 40% e 75%. A tabela apresenta o número de regras obtidas em cada faixa de eficácia: 40% a 60%, 60% a 65%, 65% a 70% e 70% a 75%. De forma geral, os experimentos com o RC retornaram resultados superiores aos alcançados pelo AG simples. Basta lembrar que no experimento realizado sem o RC e sem os parâmetros, a eficácia média das regras foi de 56,58%. Além disso, a incorporação da heurística baseada nos parâmetros de previsão também provocou melhorias significativas na eficácia da busca. É possível observar na Tabela 3 uma clara tendência dos experimentos que utilizaram a informação dos parâmetros obterem regras com eficácia acima de 65%: pelo menos 60 das 100 execuções conseguiram alcançar regras nessa faixa. Por outro lado, o experimento sem a informação dos parâmetros encontrou regras com eficácia inferior a 65% em 70 execuções. Essa melhoria possibilitou que regras com eficácia acima de 70% fossem encontradas. A regra de maior eficácia de nossos experimentos foi obtida no AG com recurso compartilhado e obteve um desempenho de 70,84%. Ela foi encontrada utilizando os parâmetros domínio da vizinhança e propagação da atividade. O código em hexadecimal dessa regra é: 020D0311- 0C191449- 230F4B1D- 85D35577- 114900D5- 91AF5CB7- 56151B1D- 91FFF1FF- 4320100B- 2319357F- 2F5C6777- 8E5F5BBB- 126E13F6- 5639718F- 076B7CF7- CED75777. Posteriormente, essa regra foi submetida a testes mais extensos e foi possível comprovar que a sua eficácia é superior à da melhor regra em [11].

TABELA II  
FAIXAS DOS PARÂMETROS DE PREVISÃO NAS REGRAS  
OBTIDAS NA TCD 2D

	$s$	$dm$	$pa$
Regras nulas	0.42 a 0.48	0.79 a 0.84	0.11 a 0.16
Regras ciclo2	0.25 a 0.48	0.22 a 0.53	0.25 a 0.38

## VII. CONCLUSÕES

A utilização de ACs bidimensionais com vizinhança de Moore para a resolução da tarefa de classificação da densidade retornou uma eficácia média inferior aos resultados obtidos pela versão unidimensional com raio 3, como já havia sido relatado no trabalho de Morales e colaboradores [11]. A melhor regra conhecida para o caso unidimensional tem uma eficácia de 86%, enquanto a melhor regra que obtivemos no caso bidimensional (e que desconhecemos resultado melhor) tem uma eficácia de pouco mais de 70%. Isso nos leva a acreditar que o ambiente evolutivo para a busca bidimensional ainda pode ser aperfeiçoado. No entanto, foi possível constatar nos experimentos relatados que a incorporação do recurso compartilhado e a heurística baseada nos parâmetros permitiram melhorias significativas no contexto bidimensional. A técnica de recurso compartilhado resultou no ambiente com os melhores resultados, embora exija um tempo de processamento maior (aproximadamente 10 vezes). A incorporação da heurística dos parâmetros, em conjunto com a técnica de recurso compartilhado, permitiu a obtenção de uma regra com 70,84%. Esta regra superou claramente a melhor regra publicada em [11], cuja eficácia é de 69%.

## REFERENCES

- [1] Binder, P.M. (1993) A Phase Diagram for Elementary Cellular Automata. *Complex Systems*, v.7, p.241–247.
- [2] Culik II, K., Hurd, L. e Yu, S. (1990) Computation Aspects of Cellular Automata. *Physica D*, 45:357-378.
- [3] Goldberg, D.E. (1989). *Genetic algorithm in search, optimization and machine learning*. Addison-Wesley.
- [4] Holland, J. (1975) *Adaptation in natural and artificial systems*. University of Michigan Press, Ann Arbor.
- [5] Juillé, H. e Pollack, J.B. (1998) Coevolving the “Ideal” Trainer: Application to the Discovery of Cellular Automata In: *Proc. of Genetic Programming Conference*. Madison, 3.
- [6] Land, M. e Belew, R. (1995) No Perfect Two-State Cellular Automata for Density Classification Exists. *Physical Review Letters*, 74(25):5148.
- [7] Langton, C. (1990) Computation at the Edge of Chaos: Phase Transitions and Emergent Computation. *Physica D*, 42:12-37.
- [8] Li, W. e Packard, N. (1990) The Structure of Elementary Cellular Automata Rule Space. *Complex Systems*, 4:281-297.
- [9] Mitchell, M.; Hraber, P. e Crutchfield, J. (1994) Evolving Cellular Automata to Perform Computations: Mechanisms and Impediments. *Physica D*, 75:361-391.
- [10] Mitchell, M. (1996) Computation in Cellular Automata: A Selected Review. In: *Nonstandard Computation*. Weinheim: VCH Verlagsgesellschaft.
- [11] Morales, F., Crutchfield, J., Mitchell, M. (2000) “Evolving two-dimensional cellular automata to perform density classification: a report on work in progress”. *Parallel Computing*, 27:571-585.

- [12] Oliveira, G.; de Oliveira, P.; Omar, N. (2000) "Evolving Solutions of the Density Classification Task in 1D Cellular Automata, Guided by Parameters that Estimate their Dynamics". *Proc. of Artificial Life VII*, p. 428 – 436.
- [13] Oliveira, G.M.B.; de Oliveira, P.P.B.; Omar, N. (2001). Definition and Applications of a Five-Parameter Characterization of One-Dimensional Cellular Automata Rule Space, *Artificial Life*, 7(3), MIT Press, p. 277-301.
- [14] Oliveira, G.M.B. (2003). Autômatos Celulares: aspectos dinâmicos e computacionais. *III Jornada de Atualização em Inteligência Artificial (MCIA)*, pp. 297-345. Sociedade Brasileira de Computação.
- [15] Oliveira, G.M.B. e Siqueira, S.R.S. (2005). Parameter Characterization of Two-Dimensional Cellular Automata Rule Space. Artigo submetido.
- [16] Packard, N. (1988) Adaptation toward the Edge of Chaos. In: *Dynamic Patterns in Complex Systems*. World Scientific, Singapore, p.293-301.
- [17] Werfel, J., Mitchell, M., Crutchfield, J. (2000) "Resource Sharing and Coevolution in Evolving Cellular Automata". *IEEE Transactions on Evolutionary Computation*, 4(4):388-393.
- [18] Wolfram, S. (1984) Computation Theory of Cellular Automata – In: *Communication in Mathematical Physics*, 96:15-57.