

ALGORITMO EVOLUTIVO APLICADO À SÍNTESE E OTIMIZAÇÃO DE CIRCUITOS BASEADOS EM QCA

Luiz Felipe V. Caires, Thiago Ferreira Noronha, Omar Paranaíba Vilela Neto

Departamento de Ciência da Computação
Universidade Federal de Minas Gerais (UFMG)
Belo Horizonte - MG - Brasil
{lfelipe, tfn, omar}@dcc.ufmg.br

RESUMO

A miniaturização do transistor permitiu construir circuitos integrados cada vez menores e com maior poder de processamento. Porém, o tamanho do transistor está próximo do seu limite físico. Uma das alternativas à esta tecnologia são os circuitos baseados em Autômatos Celulares com Pontos Quânticos (QCA). Uma das dificuldades encontradas na construção de circuitos baseados em QCA é a carência de métodos automáticos e eficientes para organizar as células em circuitos de forma a obter a lógica correta e que maximizem a polarização final das saídas. Neste trabalho é definido o problema de Síntese de Circuitos Robustos Baseados em QCA (SRQCA) e proposta uma solução baseada em Algoritmos Genéticos. Para verificar a eficiência do algoritmo em sintetizar circuitos de boa qualidade é utilizado um grupo de instâncias inspiradas em portas lógicas conhecidas da literatura. O algoritmo apresentou circuitos mais robustos que os conhecidos na literatura (que foram desenvolvidos manualmente).

PALAVRAS CHAVE. Algoritmos Genéticos, Nanocomputação, Metaheurística, QCA, Hardware Evolucionário

Metaheurística; Otimização Combinatória

ABSTRACT

The miniaturization of transistor allowed to build integrated circuits with smaller size and more processing power. However, the size of the transistor is close to its physical limits. One alternative to the conventional CMOS technology are circuits with Quantum Dots Cellular Automata (QCA). One of the difficulties found in the synthesis of circuits based on QCA is the lack of automatic and efficient methods for organizing the cells in the circuits in order to obtain the correct logic and to maximize the polarization of the output cell. This work defines the Problem of Robust Circuit Synthesis Based on QCA (SRQCA) and implements a solution based on Genetic Algorithm. To verify the efficiency of the algorithm in synthesizing robust circuits, a group of well known logic gates instances were used. The proposed algorithm found circuits more robust than those known in literature.

KEYWORDS: Genetic Algorithm, Nanocomputers, Metaheuristic, QCA, Evolvable Hardware

1. Introdução

O Autômato Celular com Pontos Quânticos (QCA - do inglês *Quantum-Dot Cellular Automata*) é um paradigma proposto por Tougaw and Lent (1994). Diferente das tecnologias tradicionais, que utilizam o fluxo de corrente elétrica para codificação das informações, a tecnologia baseada em QCA codifica as informações através da polarização das células que compõe um circuito (Lent and Tougaw, 1997).

Uma célula de QCA é formada por quatro pontos quânticos, que são regiões fixas no espaço onde os elétrons podem estar localizados. As células de QCA são construídas de forma a não permitir movimentações de elétrons para fora delas. A interação coulombiana (repulsão) existente entre dois elétrons tende a posicioná-los nas diagonais. Portanto, em uma célula, podem haver dois estados equivalentes de energia, dependendo da polarização das cargas. Os dois estados de polarização são representados como $P = +1$ e $P = -1$ e são utilizados para representar os valores binários 1 e 0, respectivamente. Neste trabalho, os valores binários 1 (verdadeiro) e 0 (falso) constantes na tabela verdade são considerados como $+1$ e -1 respectivamente, adequando as polarizações dos circuitos em QCA à lógica digital.

A polarização indica a probabilidade de se encontrar elétrons em posições específicas da célula, o valor é determinado pelas interações que ocorrem entre as células de sua vizinhança. Quanto mais próxima a polarização estiver do valor máximo (-1 ou $+1$), maior a probabilidade de se encontrar a célula em um estado distinto de polarização. A Figura 1 ilustra estes dois estados de polarização.

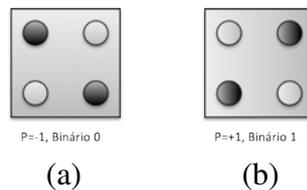


Figura 1: Possíveis polarizações de duas células de QCA isoladas, (a) representa a codificação binária 0 e (b) a codificação binária 1.

Quando uma célula com polarização fixa é posicionada próxima a uma célula com polarização livre, devido a interação existente entre os elétrons das duas células, a célula livre apresentará a mesma polarização da célula fixa. Na Figura 2 (a) é possível observar que as células (c_1 e c_2) possuem polarizações distintas, considerando que estão isoladas. O resultado da aproximação destas células pode ser visto em (b). Na ilustração é possível verificar que a interação coulombiana resultou na reorganização dos elétrons em c_2 . A região destacada (b) entre os pontos quânticos de c_1 e c_2 indica a repulsão na interação coulombiana resultante da aproximação das células.

Um fio de QCA pode ser construído posicionando células lado a lado. A Figura 3 ilustra um fio utilizando 5 células. A polarização da primeira célula c_1 foi fixada em $+1$ e as demais foram deixadas livres. Devido a interação existente entre as células, c_2 assumirá a mesma polarização da célula c_1 . Uma vez polarizada, c_2 influenciará c_3 , que assumirá a mesma polarização de sua vizinha. Estas interações ocorrem em sequência

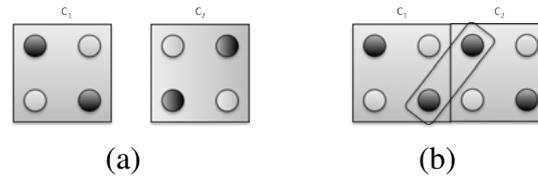


Figura 2: (a) Ilustração da interação coulombiana entre duas células c_1 e c_2 e (b) o resultado após a aproximação destas células, mantendo a polarização de c_1 fixa.

até que a última célula c_5 apresentará a mesma polarização que c_1 .

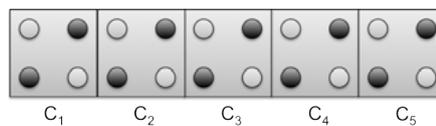


Figura 3: Ilustração de um fio utilizando 5 células de QCA.

Uma célula de QCA possui em seu interior uma barreira de tunelamento entre os pontos quânticos. Quanto menor a energia desta barreira, maior a probabilidade dos elétrons atravessá-la de um ponto quântico para outro. No valor mínimo de energia, a barreira não existe e desta forma a célula não assume nenhuma polarização, já que os elétrons não têm uma posição bem definida. Neste caso diz-se que a célula está despolarizada. Por outro lado, quando a energia está em seu valor máximo, a probabilidade dos elétrons atravessá-la será nula, desta forma os elétrons permanecem em seus pontos quânticos, definindo um dos possíveis estados de polarização. Desta forma, conforme a variação na energia da barreira de tunelamento, pode-se estabelecer quatro fases distintas de *clock*.

A síntese de circuitos lógicos utilizando CMOS utiliza um sinal que controla o tempo em que os dados são transferidos dentro do circuito, conhecido como *clock*. O *clock* nos circuitos de QCA funciona de forma ligeiramente diferente ao da tecnologia CMOS. Um ciclo de *clock* nos circuitos de QCA possui quatro fases denominados *Switch*, *Hold*, *Release* e *Relax*. As fases estão associadas ao nível de energia da barreira de tunelamento (Lent and Tougaw, 1997). Nas fases *Switch* e *Release* há uma variação na energia da barreira de tunelamento, já nas fases *Hold* e *Relax* a energia é mantida em seu valor mínimo e máximo respectivamente.

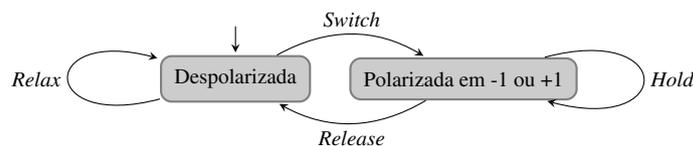


Figura 4: Diagrama de transição de estados de uma célula de QCA

A Figura 4 contém o diagrama de transição dos estados de uma célula de QCA. Neste diagrama estão representados o estado polarizado em -1 e $+1$ e o estado despo-

larizado. As fases de *clock* são representadas por transições (arcos) no diagrama. Inicialmente a célula está despolarizada ou seja, sua barreira de tunelamento possui baixa energia. Na fase Switch a energia da barreira é elevada progressivamente e os elétrons das células começam a se polarizar conforme a influência de células vizinhas. Ao fim desta fase a energia da barreira está em seu valor máximo e a célula assume um estado de polarização. Em seguida, na fase Hold, a barreira de tunelamento permanece elevada, impossibilitando a mudança de polarização. Na fase seguinte, Release, a energia da barreira é reduzida progressivamente e a célula assume um estado despolarizado. Por fim, a fase de *clock* Relax, a energia da barreira é mantida baixa e as células permanecem despolarizadas, até que um novo ciclo de *clock* se inicie.

As células de um circuito de QCA podem ser particionadas em quatro zonas distintas de *clock*, que variam da zona 0 à 3. A zona de *clock* é um recurso utilizado para garantir que os ciclos de *clock* das células, de uma mesma zona, sejam iniciados simultaneamente. O início do ciclo variará de acordo com a zona de *clock* e, o intervalo entre o início de um ciclo e outro, será o tempo de uma fase *clock*. O primeiro ciclo a ser iniciado será o da zona de *clock* 0. Em seguida, após se passar uma fase de *clock*, será iniciado o ciclo da zona 1. Após se passar mais uma fase de *clock*, será iniciado o ciclo da zona 2 e por fim, após 3 fases de *clock* será iniciado o ciclo da zona 3. Desta forma, as zonas de *clock* sempre estarão em fases distintas no ciclo de *clock*.

A topologia de um circuito QCA pode ser definido como um conjunto posições para inserção de células de QCA, cada célula em sua respectiva zona de *clock*. As células em um circuito de QCA podem ser basicamente de quatro tipos: (i) básica, (ii) de entrada, (iii) de saída e (iv) fixa. Os tipos estão relacionados à capacidade da célula em ter sua polarização definida pela influência de células vizinhas ou em ter sua polarização definida arbitrariamente .

A célula do tipo básica possui sua polarização definida pela influência das células de sua vizinhança, conforme a variação das fases de *clock*. A célula de saída é utilizada como um terminal de um circuito e seu funcionamento é o mesmo da célula básica. A partir desta célula são realizadas as conexões com o ambiente externo (e.g. os demais circuitos de um sistema digital). A polarização da célula de entrada é definida de acordo com as variações que ocorrem nas entradas da tabela verdade e são fixas durante um ciclo de *clock*. São estas células que iniciam as interações e usualmente são posicionadas nas extremidades de um circuito QCA. A polarização das células fixas são definidas previamente em -1 ou $+1$ e não se alteram ao longo dos ciclos de *clock* que ocorrem no circuito. Estas células são tipicamente utilizadas para aumentar influência de determinadas polarizações em partes específicas do circuito.

O circuito QCA que realiza a operação lógica de um inversor é ilustrado na Figura 5. As células de tonalidade escura são as de entrada e saída e as demais são células intermediárias. O destaque na figura ilustra a interação coulombiana ocorrida entre células que estão posicionadas na diagonal. Desta forma é possível observar que a polarização da célula de saída é inversa à de entrada.

A Figura 6 ilustra a topologia de um circuito QCA conhecido como *Majority Gate* de três entrada Amlani et al. (1999). As células de tonalidade escura são as três entradas

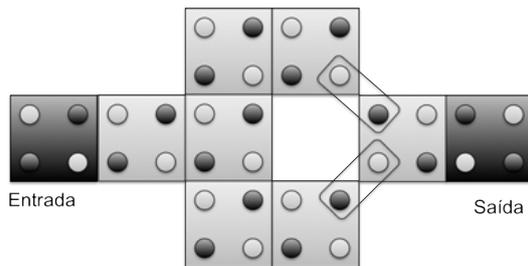


Figura 5: Circuito do Inversor

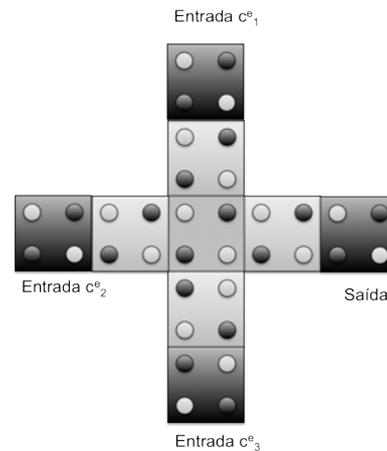


Figura 6: Circuito Majority Gate em QCA

(c_1^e, c_2^e e c_3^e) e a célula de saída. A polarização da célula de saída será aquela que mais ocorrer nas entradas. Na ilustração, as células c_1^e e c_2^e possuem polarização +1 e a célula c_3^e a polarização -1. A célula central (rachurada) sofrerá influência das três entradas. Porém, as entradas com polarização +1 estão em maior quantidade, estas exercem maior influência na célula central. A polarização +1 será propagada em direção a célula de saída.

As topologias dos circuitos de QCA que implementam as operações de E lógico (*AND*) e Ou lógico (*OR*) podem ser obtidas fixando uma das células de entrada na polarização -1 ou +1 respectivamente. Desta forma, no circuito *AND*, a saída +1 somente ocorrerá caso as duas outras entradas contenham polarização +1. Já no circuito *OR*, a célula de saída apresentará a polarização -1 somente se as duas entradas restantes tiverem as polarizações -1.

Atualmente existe uma carência de métodos automáticos para organizar as células em circuitos que desempenham a lógica esperada. Neste trabalho propõe-se desenvolver uma heurística baseada em Algoritmos Genéticos (Michalewicz, 1992), que sintetize circuitos que implementam corretamente a lógica esperada e com maior robustez.

Os próximos capítulos serão dedicado à definição formal do problema da síntese de circuitos robustos baseados em QCA (SRQCA), apresentação da solução proposta e os experimentos e resultados computacionais.

2. Trabalhos Relacionados

As pesquisas em nanotecnologia avançam e se tornam imprescindíveis no desenvolvimento e aprimoramento de novas tecnologias (Bohr, 2002). É possível verificar um volume cada vez maior de trabalhos e esforços dedicados à área, como os encontrados em Antonelli et al. (2004); Blair et al. (2010); Mahdavi et al. (2011).

No trabalho realizado por Vilela Neto et al. (2007), foi desenvolvido um método para sintetização automática de circuitos baseados em QCA que emprega uma solução baseada em Algoritmo Genético (Michalewicz, 1992). O método busca circuitos que

desempenham a lógica de acordo com uma tabela verdade minimizando o número de células empregadas no circuito.

Outros trabalhos que sintetizam circuitos mais complexos através da combinação de circuitos mais simples, tal como em Snider et al. (1999), que utilizam circuitos *Majority Gate* para sintetizar circuitos maiores. Circuitos que são desenvolvidos manualmente podem ser encontrados em Wang et al. (2003), que apresentam uma topologia de memória RAM baseado em QCA, e em Cho and Swartzlander (2007), que sintetizam um somador completo de 4 bits.

Algoritmos para síntese de sistemas nanoeletrônicos com maior robustez e tolerante a falhas, utilizando métodos baseados em Redes Neurais podem ser encontrados em (van Roermund and Hoekstra, 2000; Rouw et al., 2001; Schmid and Leblebici, 2004; Oya et al., 2005). Já algoritmos para otimização de circuitos nanoeletrônicos podem ser encontrados em Klimeck et al. (1999); Stoica et al. (1999).

3. Modelagem do Problema de Otimização

Como foi descrito anteriormente, um circuito baseado em QCA é construído através do posicionamento das células de maneira a aproveitar a interação existente entre elas.

Como sugerido por Vilela Neto et al. (2007), a área destinada ao circuito foi limitada a uma matriz de tamanho $n \times m$ células de QCA, que delimitam as possíveis posições em que uma célula pode ser posicionada. Na Figura 7 o grid possui 8×8 posições, cada ponto de cruzamento (indicado por círculos) representa o centro onde as células de QCA podem ser posicionadas. Neste exemplo há duas células de entrada (em cinza) e uma de saída (preto), o restante das posições disponíveis para fixação das células estão enumeradas (1 a 35). Neste trabalho, cada célula ocupa 4 posições no *grid*, permitindo uma maior flexibilidade para o posicionamento delas no circuito. É importante ressaltar que uma célula não pode ser posicionada sobre outra, característica de circuitos bidimensionais. No exemplo da Figura 7, ao utilizar a posição 1 para posicionar uma célula, não se pode fixar outras células nas posições 2, 4 ou 5.

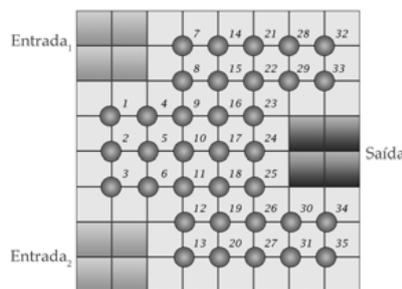


Figura 7: Ilustração do grid de um circuito com duas entradas uma saída. Cada círculo representa o centro onde as células podem ser posicionadas, ocupando quatro células da matriz.

Nos circuitos que podem ser encontradas na literatura, como em Wang et al. (2003); Vilela Neto et al. (2007); Cho and Swartzlander (2007), as células são sempre

posicionadas próximas umas das outras. A menor proximidade tende aumentar a intensidade das interações que ocorrem entre as células, minimizando possíveis interferências que atenuariam ou alterariam a polarização esperada, corrompendo as informações transmitidas pelo circuito. A robustez está ligada à capacidade de um circuito apresentar as polarizações máximas, de acordo com a lógica esperada.

Para garantir que as células serão posicionadas próximas umas das outras, todo circuito deve respeitar o princípio da conectividade. Um circuito é dito conexo quando é possível chegar às células de saída, partindo de qualquer célula de entrada, passando por células vizinhas. A vizinhança de uma célula são todas as outras imediatamente adjacentes, incluindo as da diagonal. No exemplo ilustrado pela Figura 7, se houver uma célula na posição 10, a vizinhança será composta pelas células que estarão nas posições 1, 2, 3, 8, 12, 15, 19, 22, 23, 24, 25 ou 26

O Problema de Síntese de Circuitos robustos baseados em QCA (SRQCA) pode ser definido da seguinte forma: dada uma tabela verdade $T = \{t_0, t_1, \dots, t_\alpha\}$, onde $t_i = \langle l_i, o_i \rangle$ é a i -ésima linha da tabela verdade, tal que l_i é o conjunto de valores de entrada e $o_i \in \{-1, +1\}$ é o valor esperado da saída. Dados também uma matriz $n \times m$ que define o conjunto $S = \{s_1, s_2, \dots, s_{n.m}\}$ de posições para fixação das células. O objetivo do problema é encontrar um subconjunto de posições $K \subseteq S$, para inserção de células básicas, e suas respectivas zonas de *clock* que defina a topologia de um circuito bidimensional e conexo que maximize a polarização das células de saída do circuito, de acordo com os valores esperados da tabela verdade T .

4. Solução Proposta

Neste trabalho foi implementado o algoritmo GA-QCA, que é baseado no Algoritmo Genético (GA, do inglês *Genetic Algorithm*) proposto por Holland (1975). O GA é uma metaheurística difundida na literatura que utiliza busca adaptativa para obtenção de soluções de boa qualidade. O algoritmo é baseado no princípio da genética e evolução das espécies, em que os indivíduos mais adaptados de uma população tendem a perpetuarem seus códigos genéticos para as próximas gerações, através de descendentes.

A população do algoritmo é formada por cromossomos. O cromossomo é uma estrutura de dados capaz de representar qualquer solução de um problema. Neste trabalho, o cromossomo é dividido em duas partes: a primeira parte, topologia, é utilizada para definir a ordem em que as células são inseridas em determinadas posições do *grid*. Esta parte do cromossomo é um vetor com uma permutação das posições onde uma célula de QCA pode ser fixada. A disposição em que os elementos aparecem no vetor estabelece uma ordem de prioridade em que as células são inseridas no circuito.

A segunda do cromossomo, zona de *clock*, é um vetor binário com a mesma dimensão do anterior. Os índices do vetor são as posições onde uma célula de QCA pode ser inserida. No vetor, valor 0 indica que a célula fixada, na posição indicada pelo índice, estará na mesma zona de *clock* de sua vizinhança. O valor 1 indica que a célula estará numa zona a frente de sua vizinhança. A Figura 8 ilustra um cromossomo com z genes. É possível verificar a divisão do cromossomo, contendo uma permutação das z posições e o vetor binário de zona de *clock*.

O procedimento de decodificação do cromossomo, para a geração de um circuito

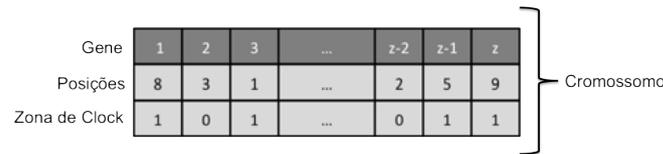


Figura 8: Ilustração da representação do cromossomo com z genes.

completo, é um processo iterativo de inserção das células, baseado na ordem em que as posições estão dispostas na parte do cromossomo de topologia. Como as células de entrada e saída são posicionadas previamente, são ocupadas primeiramente as posições nas proximidades das células de entrada, até que todos os genes sejam decodificados, gerando assim uma topologia completa. No exemplo da Figura 7, a primeira célula poderia ser inserida nas posições 1, 3, 4, 6, 7, 8, 9, 11, 12 ou 13, respeitando a restrição de conectividade.

Uma célula somente poderá ser fixada se houver pelo menos outra em sua vizinhança, excluindo as células do tipo fixa ou de saída. Partindo do primeiro gene, deve ser verificada a viabilidade de inserção da célula na posição indicada. Caso não seja possível concluir a inserção é realizada a tentativa com o gene seguinte. Assim que uma célula é fixada em determinada posição, é definida também sua zona de *clock* (segundo o valor binário do seu respectivo cromossomo), e a iteração retorna ao primeiro gene. Desta forma, pode-se realizar novas tentativas em posições que possuem uma maior prioridade, de acordo com a ordem estabelecida, mas que não foram utilizadas em função de alguma inviabilidade anterior. O procedimento é repetido até que o circuito se torne conexo.

O *fitness* é um valor atribuído a cada indivíduo da população e indica o custo da solução segundo a função objetivo adotada. Neste trabalho, o *fitness* é calculado da seguinte forma: dado o conjunto de amostras $\pi_i = \{\pi_i^0, \pi_i^1, \dots, \pi_i^q\}$ onde π_i^j é j -ésima polarização amostrada na célula de saída no instante em que era testada a i -ésima entrada na tabela verdade T , e dado o conjunto de erros $\varepsilon_i = \{\varepsilon_i^0, \varepsilon_i^1, \dots, \varepsilon_i^q\}$ onde $\varepsilon_i^j = \phi(|o_i - \pi_i^j|)$ é o custo do erro obtido entre o valor esperado de polarização o_i e a polarização amostrada π_i^j , em que ϕ é uma função que determina o custo do erro. A função de custo utilizada neste trabalho (Equação 1) é contínua definida por partes e foi baseada em Buriol et al. (2005). É acrescido ao erro um custo que varia à medida que o valor da polarização se distancia do valor desejado, aumentando quando se torna o inverso do esperado. Esta abordagem permite segmentar o erro apresentado pelas células de saídas de um circuito.

$$\phi(\varphi) = \begin{cases} \varphi, & \varphi \in [0, 1/10[\\ 3 \times \varphi - 1/5, & \varphi \in [1/10, 1/2[\\ 10 \times \varphi - 37/10, & \varphi \in [1/2, 9/10[\\ 70 \times \varphi - 577/10, & \varphi \in [9/10, 1[\\ 500 \times \varphi - 487/10, & \varphi \in [1, 11/10[\\ 5000 \times \varphi - 54377/10, & \varphi \in [11/10, 2 \end{cases} \quad (1)$$

Em determinados circuitos um tipo de saída pode ocorrer mais que outro, como a tabela verdade de um circuito AND. Nesta topologia, do circuito AND, se a célula de saída apresentar somente a polarização -1 (curto-circuito), a topologia deste circuito seria bem avaliada, já que este possui 75% das saídas esperadas, direcionando algoritmo rapidamente à um mínimo local.

Para evitar que o algoritmo convirja para soluções exemplificadas anteriormente, é realizado uma ponderação de acordo com a quantidade de ocorrências que determinado valor de saída apresenta em relação a outro. O *fitness* de um indivíduo é dado por ξ (Equação 2), em que $O^{+1} = \{i | o_i = +1\}$ são os índices das linhas na tabela verdade cuja polarização de saída é $+1$. De forma análoga, $O^{-1} = \{i | o_i = -1\}$ é o conjunto dos índices das linhas na tabela verdade cuja polarização de saída é -1 .

$$\xi = \frac{1}{2} \left(\frac{\sum_{i \in O^{-1}} \sum_{j=0}^q \varepsilon_i^j}{|O^{-1}|^q} + \frac{\sum_{i \in O^{+1}} \sum_{j=0}^q \varepsilon_i^j}{|O^{+1}|^q} \right) \quad (2)$$

No procedimento de seleção são escolhidos os indivíduos que serão utilizados para geração de descendentes. Cada indivíduo possui uma probabilidade de ser selecionado, que é proporcional ao seu *fitness*. Neste trabalho foi adotado o método baseado em *Ranking linear*.

O cruzamento é um procedimento em que novos descendentes são criados a partir da combinação de elementos presentes nos cromossomos de dois outros. Neste trabalho foi adotado operador de cruzamento *PMX* e *OX* (Michalewicz, 1992).

A mutação é um procedimento utilizado para inserir novos elementos na população, através da alteração aleatória em partes do cromossomo dos indivíduos. No algoritmo proposto, foi implementado o operador de mutação *Reciprocal Exchange* (Michalewicz, 1992).

5. Experimentos e Resultados Computacionais

Neste trabalho usamos um simulador para avaliar os circuitos de QCA. O *QCA-Designer* é um simulador que implementa dois tipos de simulações para determinar a polarização das células de um circuito. A primeira simulação emprega cálculos baseados em lógica digital, que são mais simplificados e que demandam menos processamento, já a segunda é baseada em mecânica quântica, que demanda maior processamento, porém é mais precisa (Walus et al., 2011).

Para verificar a eficiência do algoritmo proposto na seção anterior, foram utilizadas instâncias clássicas baseadas em operadores lógicos comumente encontrados na literatura. Os circuitos gerados pelo GA-QCA são comparados com as topologias de Amlani et al. (1999), que foram desenvolvidas manualmente.

Neste trabalho foram testadas 4 instâncias. Três delas foram baseadas na porta lógica OR de duas entradas, em que duas destas possuem o *grid* de tamanho 10×10 e uma de tamanho 12×12 . Uma das instâncias, baseada na porta OR (*I_OR_12_FIXED*), foi utilizada uma célula com polarização fixa em 1, da mesma forma que ocorre no circuito feito por Amlani et al. (1999). Esta é a única forma de sintetizar uma porta OR encontrada na literatura. Em outra instância, baseada na porta OR (*I_OR_12_FREE*),

nenhuma célula foi fixada, deixando a cargo do algoritmo definir a topologia mais adequada. A última instância (I_NAND_12_FREE) foi baseada na porta NAND de duas entradas, também com um grid de tamanho 12×12 e uma célula fixa em -1 .

O algoritmo GA-QCA foi implementado em linguagem C++ e foram executados em um computador Intel de 2.8 GHz, com 8 GB de memória RAM e sistema operacional Linux. Em todas as execuções do GA foram adotado os seguintes parâmetros: 40% da população é gerada a partir do operador de Mutação e 40 % através do operador Crossover e o restante. Os 20% melhores cromossomos são copiados para geração seguinte. O critério de parada foi fixado em 250 gerações e população com 250 indivíduos.

O algoritmo foi executado por 10 vezes para cada instância, variando a semente do gerador de números aleatórios e o resultados podem ser conferidos na Tabela 5. A tabela contém a comparação dos resultados obtidos com as execuções do algoritmo proposto em relação aos circuitos da literatura, desenvolvidos manualmente. A primeira coluna da tabela contém o nome das instâncias utilizadas, a coluna seguinte contém o número de posições disponíveis para a fixação das células. A terceira e quarta colunas contém, respectivamente, o número de células e a polarização média observada nos circuitos da literatura. Os campos que contém “-”, indicam que não foram encontradas topologias na literatura para comparação. As quatro últimas colunas são dedicadas ao algoritmo proposto. Na quinta está quantidade média de células utilizadas nos circuitos obtidos pelas execuções do GA-QCA. Na sexta está a polarização média observada pelos circuitos obtidos pelas execuções do algoritmo. Em seguida, é mostrada a variação percentual e o tempo médio gasto, em minutos, para cada execução do algoritmo. O valor da variação percentual é obtido por $\frac{(POL_MED - POL_MED_{LIT}) * 100}{POL_MED_{LIT}}$, em que POL_MED é o valor médio das polarizações das soluções obtidas nas execuções do algoritmo e POL_MED_{LIT} é o valor da polarização da solução conhecida da literatura.

INSTANCIA	N_POS	Literatura		Algoritmo Proposto			
		NUM_CEL	POL_MED	NUM_CEL	POL_MED	VP	TEMPO
I_OR_12_FREE	132	-	-	15	0,975	-	275
I_OR_12_FIXED	128	12	0,954	14	0,96	0,63	320
I_OR_10_FIXED	84	11	0,954	16	0,961	0,73	230
I_NAND_12_FIXED	128	11	0,775	20	0,99	27,74	373

Tabela 1: Comparação dos resultados obtidos pelo Algoritmo Proposto, em relação aos circuitos desenvolvidos manualmente, em Amlani et al. (1999). Os campos que contém “-”, indicam que não foram encontradas topologias na literatura para comparação.

É possível observar que o algoritmo proposto foi capaz de gerar circuitos válidos para todas as instâncias e com maior polarização nas células de saída que as sintetizados manualmente. No caso da instância I_OR_12_FREE, não há na literatura circuitos OR sem célula fixa, indicando que a topologia alcançada é inédita, já que as topologias deste tipo de circuito utilizam *Majority Gate* de três entradas, uma das entradas fixa em $+1$.

O resultado obtidos com a instância I_NAND_12_FREE apresentou a maior di-

ferença entre a polarização média do circuito encontrado na literatura e o obtido com algoritmo proposto, cerca de 27%. Embora o número de células não tenha sido considerada na função objetivo do problema, para a instância I_OR_12_FREE os circuitos encontrados pelas execuções algoritmo proposto apresentaram uma quantidade próxima da literatura.

6. Conclusão

Os experimentos computacionais realizados mostraram que o algoritmo apresentou resultados melhores que aqueles obtidos com os circuitos desenvolvidos manualmente, encontrados na literatura, com relação a polarização das células de saída. O algoritmo foi capaz de encontrar uma topologia para o circuito OR, que não utiliza uma célula com polarização fixa.

Trabalhos futuros podem ser direcionados no estudo de modelos para solução do problema real como, por exemplo, uma abordagem multi-objetivo que possa maximizar as polarizações e reduzir número de células utilizadas em um circuito.

Agradecimentos

Este trabalho foi parcialmente apoiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), pela Fundação de Apoio a Pesquisa do Estado de Minas Gerais (FAPEMIG) e pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES).

Referências

- I. Amlani, A.O. Orlov, G. Toth, G.H. Bernstein, C.S. Lent, and G.L. Snider. Digital logic gate using quantum-dot cellular automata. *Science*, 284(5412):289–291, 1999.
- D.A. Antonelli, D.Z. Chen, T.J. Dysart, X.S. Hu, A.B. Kahng, P.M. Kogge, R.C. Murphy, and M.T. Niemier. Quantum-dot cellular automata (qca) circuit partitioning: problem modeling and solutions. In *Proceedings of the 41st annual Design Automation Conference*, pages 363–368. ACM, 2004.
- E.P. Blair, E. Yost, and C.S. Lent. Power dissipation in clocking wires for clocked molecular quantum-dot cellular automata. *Journal of computational electronics*, 9(1):49–55, 2010.
- M.T. Bohr. Nanotechnology goals and challenges for electronic applications. *Nanotechnology, IEEE Transactions on*, 1(1):56–62, 2002.
- LS Buriol, MGC Resende, C.C. Ribeiro, and M. Thorup. A hybrid genetic algorithm for the weight setting problem in ospf/is-is routing. *Networks*, 46(1):36–56, 2005.
- H. Cho and E.E. Swartzlander. Adder designs and analyses for quantum-dot cellular automata. *Nanotechnology, IEEE Transactions on*, 6(3):374–383, 2007.
- J.H. Holland. *Adaptation in natural and artificial systems*. Ann Arbor: University of Michigan Press, 2, 1975.
- G. Klimeck, C.H. Salazar-Lazaro, A. Stoica, and T. Cwik. “genetically engineered” nanoelectronics. In *Evolvable Hardware, 1999. Proceedings of the First NASA/DoD Workshop on*, pages 247–248. IEEE, 1999.
- C.S. Lent and P.D. Tougaw. A device architecture for computing with quantum dots. *Proceedings of the IEEE*, 85(4):541–557, 1997.
- M. Mahdavi, S. Mirzakuchaki, MN Moghaddasi, and MA Amiri. Single electron fault modelling in quantum binary wire. *Micro & Nano Letters, IET*, 6(2):75–77, 2011.
- Z. Michalewicz. Genetic algorithms+ data structures= evolution program. *Artificial Intelligence, Berlin: Springer, 1992*, 1, 1992.
- T. Oya, T. Asai, Y. Amemiya, A. Schmid, and Y. Leblebici. Single-electron circuit for inhibitory spiking neural network with fault-tolerant architecture. In *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on*, pages 2535–2538. IEEE, 2005.

- E. Rouw, J. Hoekstra, and AHM Van Roermund. Spike correlation based learning for unsupervised neural lattice structures. In *Circuits and Systems, 2001. ISCAS 2001. The 2001 IEEE International Symposium on*, volume 3, pages 425–428. IEEE, 2001.
- A. Schmid and Y. Leblebici. Robust circuit and system design methodologies for nanometer-scale devices and single-electron transistors. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 12(11):1156–1166, 2004.
- G.L. Snider, A.O. Orlov, I. Amlani, G.H. Bernstein, C.S. Lent, J.L. Merz, and W. Porod. Quantum-dot cellular automata: Line and majority logic gate. *Japanese Journal of Applied Physics*, 38(part 1): 7227–7229, 1999.
- A. Stoica, G. Klimeck, C. Salazar-Lazaro, D. Keymeulen, and A. Thakoor. Evolutionary design of electronic devices and circuits. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, volume 2. IEEE, 1999.
- P.D. Tougaw and C.S. Lent. Logical devices implemented using quantum cellular automata. *Journal of Applied physics*, 75(3):1818–1825, 1994.
- A. van Roermund and J. Hoekstra. From nanotechnology to nanoelectronic systems, from sets to neural nets. In *Circuits and Systems, 2000. Proceedings. ISCAS 2000 Geneva. The 2000 IEEE International Symposium on*, volume 1, pages 8–11. IEEE, 2000.
- O.P. Vilela Neto, M.A.C. Pacheco, and C.R.H. Barbosa. Neural network simulation and evolutionary synthesis of qca circuits. *IEEE Transactions on Computers*, pages 191–201, 2007.
- K. Walus, V. Dimitrov, G. A. Jullien, and W. C. Miller. Qcadesigner, 2011.
- Wei Wang, K. Walus, and G.A. Jullien. Quantum-dot cellular automata adders. In *Nanotechnology, 2003. IEEE-NANO 2003. 2003 Third IEEE Conference on*, volume 1, pages 461 – 464 vol.2, aug. 2003. doi: 10.1109/NANO.2003.1231818.