UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"

FACULDADE DE ENGENHARIA

CAMPUS DE ILHA SOLTEIRA

CAROLINE DOMINGUES PORTO DO NASCIMENTO BARBIERI

ALGORITMOS PARA A SÍNTESE DE CIRCUITOS REVERSÍVEIS TERNÁRIOS: ANÁLISE COMPARATIVA

Ilha Solteira
2018

UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Campus de Ilha Solteira

CAROLINE DOMINGUES PORTO DO NASCIMENTO BARBIERI

ALGORITMOS PARA A SÍNTESE DE CIRCUITOS REVERSÍVEIS TERNÁRIOS: ANÁLISE COMPARATIVA

Tese apresentada à Faculdade de Engenharia - UNESP – Campus de Ilha Solteira, como parte dos requisitos para obtenção do título de Doutora em Engenharia Elétrica.

Área de Conhecimento: Automação.

Prof. Dra. Anna Diva Plasencia Lotufo

Orientadora

Prof. Dr. Claudio Moraga

Coorientador

Ilha Solteira
2018

FICHA CATALOGRÁFICA
Desenvolvido pelo Serviço Técnico de Biblioteca e Documentação

B236a

Barbieri, Caroline Domingues Porto do Nascimento.
Algoritmos para a síntese de circuitos reversíveis ternários: análise comparativa / Caroline Domingues Porto do Nascimento Barbieri. -- Ilha Solteira: [s.n.], 2018
109 f. : il.

Tese (doutorado) - Universidade Estadual Paulista. Faculdade de Engenharia de Ilha Solteira. Área de conhecimento: Automação, 2018

Orientadora: Anna Diva Plasencia Lotufo
Coorientador: Claudio Moraga
Inclui bibliografia

1. Lógica ternária. 2. Computação quântica. 3. Síntese de circuitos reversíveis. 4. Permutação. 5. Ciclos de permutação disjuntos. 6. Algoritmo genético.

Raiane da Silva Santos
Supervisora Técnica de Seção
Seção Técnica de Referência, Atendimento ao usuário e Disseminação
Diretoria Técnica de Biblioteca e Documentação
CRB/8 - 9999

unesp UNIVERSIDADE ESTADUAL PAULISTA

Câmpus de Ilha Solteira

CERTIFICADO DE APROVAÇÃO

TÍTULO DA TESE: ALGORITMOS PARA A SINTESE DE CIRCUITOS REVERSIVEIS TERNARIOS: ANALISE COMPARATIVA

AUTORA: CAROLINE DOMINGUES PORTO DO NASCIMENTO BARBIERI
ORIENTADORA: ANNA DIVA PLASENCIA LOTUFO

Aprovada como parte das exigências para obtenção do Título de Doutora em ENGENHARIA ELÉTRICA, área: AUTOMAÇÃO pela Comissão Examinadora:

Prof. Dr. CLAUDIO MORAGA
Dept. Computer Science / Technical University of Dortmund

Prof. Dr. CARLOS ROBERTO MINUSSI
Departamento de Engenharia Elétrica / Faculdade de Engenharia de Ilha Solteira

Profa. Dra. MARA LUCIA MARTINS LOPES
Departamento de Matemática / Faculdade de Engenharia de Ilha Solteira

Prof. Dr. KENJI NOSE FILHO
Centro de Engenharia, Modelagem e Ciências Sociais Aplicadas / Universidade Federal do ABC

Profa. Dra. FERNANDA CASENO LIMA TRINDADE ARIOLI
Departamento de Sistemas de Energia Elétrica / Universidade Estadual de Campinas - UNICAMP

Ilha Solteira, 24 de agosto de 2018

AGRADECIMENTOS

A Deus, que por sua presença, sempre ouve as minhas orações, e me capacita para tudo aquilo que Ele me destina. Aos meus avós, pelo modelo de vida passado, com muito esforço e espírito de luta alçaram conquistas gloriosas, sem deixar de lado os maiores valores da vida. Acreditaram, deram apoio, e acima de tudo me ensinaram o poder da fé, elementos essências para esta conquista. Ao Samuel, a quem devo a felicidade e o amor, e que além de meu esposo, tem sido meu companheiro e grande amigo. À minha mãe Ângela Cristina Domingues Porto, que me mostrou que é preciso coragem para vencer. Ao meu Pai Wilson Barbieri. À minha querida tia Carla Cristiane Domingues Porto pelo imenso carinho e principalmente pelas importantes revisões ortográficas, que me ajudaram na escrita deste trabalho. À minha prezada orientadora Prof. Dra. Anna Diva Plasencia Lotufo, pela grande ajuda e por me estender a sua mão amiga em um dos momentos mais difíceis da minha jornada, para que este trabalho fosse realizado. Ao Prof. Dr. Carlos Roberto Minussi pelo importante amparo na realização deste trabalho. Ao meu caro coorientador Prof. Dr. Claudio Moraga pelas oportunidades que me proporcionou, abrindo a porta para que trabalhássemos em conjunto, pela disponibilidade, paciência e principalmente pelo grande apoio e amizade durante o período em que trabalhamos juntos. Ao meu amigo Jean Vitor, e aos demais, que me auxiliaram por meio da amizade e ajuda. A CNPq pelo auxílio financeiro para a realização desta pesquisa, e à CAPES pelo auxílio financeiro durante o desenvolvimento parcial no exterior. A todos que contribuíram indiretamente para a realização deste trabalho.

RESUMO

A lógica de múltiplos valores, em especial a ternária, apresenta inúmeras vantagens sobre a lógica binária em circuitos reversíveis/quânticos. A realização de funções usando a lógica reversível ternária é conhecida por requerer um menor número de linhas em comparação com a lógica reversível binária convencional. Este aspecto tem motivado as pesquisas em abordagens de síntese. A grande maioria dos métodos existentes requerem entradas adicionais, denominadas de *ancillary lines*, durante o processo de síntese, o que é dispendioso para implementação em tecnologias quânticas, quando disponíveis. Neste trabalho, foram propostas diferentes metodologias e análises comparativas para o problema da síntese de circuitos reversíveis ternários sem a adição de *ancillary lines.* A metodologia de síntese proposta, denominada de MMD *plus,* foi aplicado nos modos *backward* e *top-down* como referência a todas as 362880 possíveis funções reversíveis ternárias de 2 variáveis. Além do processamento *top-down* originário do algoritmo MMD, um processamento *bottom-up* é implementado e sua eficiência comparativa é avaliada. Por definição, as funções reversíveis ternárias são permutações. Realiza-se a decomposição das permutações em ciclos disjuntos de ordem natural, em ciclos de permutação com 3 elementos, e em transposições, para obtenção dos circuitos reversíveis ternários. Uma métrica é introduzida para mensurar a complexidade e custo dos circuitos, com base nas portas reversíveis de múltiplos valores Muthukrishnan-Stroud (MS). A eficiência do procedimento de síntese do MMD *plus* empregado aos ciclos de permutação é avaliada e comparada com a aplicação direta, isto é, sem decomposição das permutações em ciclos. Em seguida, propõem-se duas metodologias com base em Algoritmos Genéticos para o projeto de circuitos reversíveis ternários como um problema de otimização, utilizando-se, também, das portas MS. A influência de diferentes tipos de seleção e variação do tamanho dos indivíduos é avaliada em relação à quantidade de portas e custos dos circuitos resultantes da síntese.

Palavras-Chave: Lógica ternária. Computação quântica. Síntese de circuitos reversíveis. Permutação. Ciclos de permutação disjuntos. Algoritmo genético.

ABSTRACT

Multiple-valued logic, especially ternary logic, has many advantages over binary logic in reversible/quantum circuits. Performing functions using ternary reversible logic is known to require fewer rows compared to conventional binary reversible logic. This aspect has motivated the researches in synthetic approaches. The vast majority of existing methods require additional lines, called ancillary lines, during the synthesis process, which is expensive for implementation in quantum technologies, when available. In this work, were proposed different methodologies and comparative analysis for the problem of the synthesis of ternary reversible circuits without the addition of ancillary lines. The proposed synthesis methodology, called MMD plus, in backward top-down modes was applied as a reference to all 362880 possible ternary 2-variable reversible functions. In addition to the original top-down processing of the MMD algorithm, a bottom-up processing is implemented and its comparative efficiency is evaluated. By definition the ternary reversible functions are permutations. The decomposition of the permutations is performed in disjoint cycles of the natural order, in permutation cycles with 3 elements and in transpositions, to obtain the circuits. A metric is introduced to estimate the complexity or cost of circuits based on Muthukrishnan-Stroud (MS) multivalued reversible gates. The efficiency of the MMD plus synthesis employed in the permutation cycles is evaluated and compared with the direct application, without decomposition. Then, two methodologies based on Genetic Algorithms are proposed for the design of ternary reversible circuits as an optimization problem, using also the MS gates. The influence of different types of selection and variation of the individual's size is evaluated in relation to the number of gates and costs of the circuits resulting from the synthesis.

Keywords: Ternary logic. Quantum computation. Synthesis of reversible circuits. Permutation. Disjunct permutation cycles. Genetic algorithm.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Por causa da capacidade de ser aplicada em diferentes tipos de tecnologias emergentes, proporcionar baixa dissipação de calor em dispositivos eletrônicos e, principalmente, pelo potencial de implementações na computação quântica, considerada uma das áreas mais promissoras para o futuro dos sistemas computacionais, as pesquisas sobre circuitos reversíveis ganharam impulso nos últimos anos (HADJAM; MORAGA, 2014; SOEKEN; DUECK; MILLER, 2016; RICE, 2016; NAGAMANI et al., 2018).

Embora a maioria das aplicações para circuitos reversíveis seja atualmente em tecnologias emergentes, as primeiras considerações e definições para um processo computacionalmente reversível foram feitas entre as décadas de 1960 e 1980.

Se em um determinado processo a informação gerada como saída for suficiente para calcular a informação de entrada, este processo é considerado reversível, ou seja, nenhuma informação é destruída durante a realização dos cálculos (TOFFOLI, 1980). As portas lógicas convencionais, com exceção da porta NOT, não são reversíveis, pois os valores de entrada não podem ser calculados conhecendo unicamente os valores de saída. Landauer (1961) conseguiu determinar a quantidade de dissipação de calor que é perdida em cada bit de operação irreversível nas portas convencionais. Posteriormente, Bennett (1973) mostrou que computadores com a dissipação de energia tendendo a zero só são possíveis quando opera perto de seu equilíbrio termodinâmico e, além disso, mostrou que este equilíbrio pode ser conseguido por meio de componentes reversíveis. Bennett afirmou também que, para evitar a dissipação de energia, o circuito deve ser construído a partir de portas reversíveis.

É relevante ressaltar que todas as operações quânticas básicas são reversíveis e, por isso, a computação quântica é considerada a principal aplicação dos circuitos reversíveis. Os computadores quânticos têm sido considerados por importantes pesquisadores e empresas, como a IBM, como o futuro da computação, uma vez que os computadores convencionais estão atingindo seus limites físicos (MOORE, 1965; BENNETT; DIVINCENZO, 2000). Por meio da superposição de bits, estes computadores permitem o processamento paralelo intensivo de cálculos, fazendo com que problemas exponenciais não solucionáveis pela computação clássica sejam solucionados em tempo polinomial por computadores quânticos, como no caso do algoritmo de fatoração proposto por Shor (1999).

Isto posto, a síntese e otimização de circuitos reversíveis estão estritamente relacionadas à computação quântica. Além disso, os circuitos reversíveis resultantes da síntese são

usualmente mapeados para uma biblioteca de portas quânticas para avaliar o custo de implementação. Como exemplo, tem-se o modelo de decomposição de portas reversíveis em quânticas, proposto por Barenco et al. (1995). Modelos quânticos similares modelos quânticos para portas reversíveis operando sobre múltiplos valores também têm sido propostos e recebido atenção especial nos dias que correm (AHARONOV; BEN-OR, 2008; MORAGA, 2016-a; MORAGA 2016-b). Esta atenção especial se deve ao fato de que portas reversíveis trabalhando com múltiplos valores possibilitam uma codificação de informação mais compacta e eficiente. Ao passo que no domínio binário têm-se duas unidades de informação 0 e 1, no domínio de múltiplos valores é possível trabalhar com *p* unidades de informação.

A escolha de um modelo binário foi inevitável no princípio dos computadores clássicos, uma vez que os componentes básicos disponíveis (válvulas/ transistores) tinham apenas dois estados reais controláveis. No contexto quântico, esta condição é diferente. Pesquisadores têm trabalhado experimentalmente com portas quânticas com dois valores/níveis ou mais (MORAGA, 2016). O trabalho mais significativo neste contexto foi desenvolvido por Muthukrishnan e Stroud (2000), no qual reportaram a realização física de portas com *p*-níveis sob a tecnologia quântica *ion-trap*. Foi mostrado, também, que o emprego de operações com múltiplos valores proporciona maior flexibilidade no armazenamento e no processamento das informações.

No domínio de múltiplos valores, circuitos reversíveis empregando-se de três valores, 0, 1 e 2, são referidos como ternários. Logo, a quantidade de informação que pode ser armazenada por operação na computação ternária é visivelmente maior quando comparada à binária. As operações ternárias levam à redução do número de linhas na concepção de um circuito reversível, bem como a redução do número de portas quando implementadas em circuitos quânticos. Especificamente, uma determinada função reversível binária, quando codificada na forma ternária, torna-se 63% mais compacta em relação ao número de linhas no circuito quântico (KHAN, 2010). Em virtude destas características, a síntese de circuitos reversíveis ternários tem emergido como um importante problema de pesquisa nos últimos anos (KHAN; RICE, 2016; KOLE et al., 2017; KHAN et al., 2017).

Um circuito, binário ou ternário, é considerado reversível se for realizado por uma cascata acíclica de portas reversíveis livres de *fan-out*, cujas portas realizam funções bijetivas e, por conseguinte, devem conter o mesmo número de variáveis de entrada e de saída. Define-se como *fan-out*, o número máximo de entradas que podem ser alimentadas pela saída de uma porta lógica. Essas restrições evidenciam que projetos de circuitos reversíveis são significativamente mais complexos do que os circuitos digitais clássicos (irreversíveis). A

complexidade está no fato de que o espaço de busca pela solução é sempre estendido, sendo não trivial, e as propriedades de reversibilidade impostas, como *fan-out*, *feedback* e portas com o mesmo número de entradas/saídas, dificultam implementações (SASANIAN, 2012).

Portanto, definir uma metodologia de síntese eficiente permite a geração de circuitos reversíveis ternários menos complexos como viabiliza menores custos de implementação em futuras tecnologias quânticas.

Encontram-se na literatura poucos trabalhos desenvolvidos para a síntese reversível ternária quando comparados às inúmeras contribuições encontradas para síntese reversível binária, em especial os métodos livres de *ancillary lines* (YANG et al., 2005; MILLER; MASLOV; DUECK, 2006). *Ancillary lines* são variáveis de entrada extras adicionadas a uma função não reversível com a finalidade de torná-la reversível.

Miller, Maslov e Dueck (2006) propuseram uma metodologia heurística, relevante para a comunidade de múltiplos valores, explorando a síntese reversível ternária sem linhas auxiliares nos circuitos gerados. Tal método foi fundamentado no algoritmo MMD para a síntese de circuito reversível binário, desenvolvido anteriormente pelos mesmos autores (MILLER; MASLOV; DUECK, 2003). Embora não explicitamente mencionado pelos autores, cada passo do algoritmo MMD compreende uma permutação, independentes entre si, bem como qualquer permutação pode ser escrita como uma cascata de ciclos de permutação (BAUMSLAG; CHANDLER, 1968).

No domínio reversível binário, Shende et al. (2003) formalizaram esta abordagem cíclica. As especificações reversíveis foram decompostas em ciclos de permutação de tamanho 2 (ciclos com dois elementos), denominados de transposições. Posteriormente, Sasanian et al. (2009) provaram que as transposições geram redundâncias e, então, introduziram decomposições de permutações usando ciclos de tamanho 3 e, possivelmente, um ciclo adicional de tamanho 2. Entretanto, no domínio ternário, exceto pelo trabalho de Li et al. (2013), salvo engano, não se conhece outros trabalhos que apliquem a decomposição de permutação em ciclos para a síntese de circuitos reversíveis ternários. Além disso, o método construtivo proposto por tais autores realiza a adição de entradas constantes e saídas *garbages* nas especificações de entrada, gerando linhas extras na implementação e, consequentemente, acrescendo o custo dos circuitos gerados. A variável de saída extra adicionada a uma função com a finalidade de torná-la reversível é denominada de *garbage* .São apresentados dois exemplos de empregos da metodologia construtiva, o de um meio somador e de um somador completo. É importante salientar que Yang et al. (2005) comprovaram teoricamente que as

propriedades referentes a grupos de permutação podem ser aplicadas diretamente para especificações ternárias.

Neste cenário, o presente trabalho se insere. O problema de síntese de circuitos reversíveis ternários é explorado. Utilizando-se de um conjunto específico de portas reversíveis ternárias para obter novas concepções sobre os principais aspectos do algoritmo MMD ternário (MILLER; MASLOV; DUECK, 2006), propõe-se uma versão estendida denominada de MMD *plus*. Nesta versão estendida, as portas do tipo *Muthukrishnan-Stroud* (MS) *(*MUTHUKRISHNAN; STROUD; 2000) são empregadas na síntese direta às especificações reversíveis ternárias e, particularmente, às especificações decompostas na forma de ciclos de permutação de ordem natural, na forma de ciclos de permutação de tamanho 3, e em transposições. As abordagens propostas são testadas exaustivamente para todas as possíveis especificações reversíveis ternárias de duas variáveis, $m = 2$. Logo, existem 362880 especificações, isto é, $3^m!$.

Não é possível garantir que os circuitos reversíveis ternários gerados por meio da síntese direta ou aplicados às diversas formas de ciclos de permutação sejam mínimos, pois o espaço de busca por uma solução mínima, ou próxima do mínimo, aumenta exponencialmente de acordo com o número de variáveis de entrada. Em sistemas ternários, o espaço de busca se expande a uma taxa exponencial de $(3^m)!$ Consequentemente, é inviável usar qualquer método direto ou determinístico para tal finalidade.

Algoritmos evolutivos, com busca orientada a multiobjetivos, têm gerado bons resultados quando utilizados em processos de otimização e, também, na geração de circuitos reversíveis binários mínimos ou próximos do mínimo (LUKAC et al., 2003; HADJAM; MORAGA, 2014). Isso se deve ao fato de que o espaço de busca pode ser delimitado por parâmetros favoráveis à solução de determinados problemas, convergindo em uma variedade de novas soluções importantes, e em um tempo computacional aceitável.

Tais características motivaram o desenvolvimento de dois Algoritmos Genéticos adaptados para o projeto de circuitos reversíveis ternários como um problema de otimização. A aplicação de Algoritmos Genéticos para o problema de síntese já tem sido investigada (KHAN et al., 2007; KHAN, 2008; KHANOMA; KAMALB; KHANA, 2008; DEIBUK; BILOSHYTSKYI, 2015; HU; DEIBUK, 2016). No entanto, a obtenção de circuitos mínimos ou próximos do mínimo para a síntese reversível ternária permanece um desafio. O problema de encontrar a estrutura ideal é de difícil abordagem, uma vez que a estrutura de uma determinada cascata de portas reversíveis ternárias não é conhecida incialmente, e diferentes parâmetros devem ser configurados na busca por uma solução eficiente. O custo total de um

circuito reversível ternário é definido pelo somatório do custo individual de cada porta reversível ternária. Um determinado circuito com o mínimo de entradas auxiliares, de custo total e de portas é considerado eficiente. Assim sendo, apresentam-se duas abordagens práticas, que permitem encontrar uma combinação apropriada de portas reversíveis ternárias do tipo MS que realizem qualquer especificação reversível ternária de duas variáveis. O objetivo no processo de busca pela solução foi à realização de circuitos com a menor quantidade de portas possíveis em relação aos obtidos pelo MMD *plus*. Em uma segunda versão do algoritmo, propôs-se realizar a busca orientada a dois parâmetros que influenciam diretamente na qualidade dos circuitos gerados, a quantidade de portas e o custo total.

1.1 OBJETIVOS E CONTRIBUIÇÕES

O principal objetivo deste trabalho consiste no desenvolvimento de diferentes metodologias para explorar a síntese de circuitos reversíveis ternários, livres de linhas auxiliares, para todas as possíveis especificações reversíveis ternárias com duas variáveis (m = 2).

Dentre as contribuições principais destacam-se:

- Aplicar o procedimento de síntese da metodologia proposta, MMD *plus,* à teoria de grupo, especificamente a ciclos de permutação de ordem natural, a ciclos de tamanho 3 e transposições, bem como as vantagens e desvantagens da aplicação neste contexto;
- Apresentar os resultados de testes exaustivos a partir de um *benchmark* completo, contendo 362880 especificações reversíveis ternárias de duas varáveis, na aplicação direta do método MMD *plus* e também nas três diferentes formas de ciclos de permutação propostas neste trabalho;
- Apresentar os benefícios de explorar diferentes tipos de ordenamentos (lexicográfico, por distância *Hamming* e por peso *Hamming*) para os vetores de entrada do algoritmo MMD *plus*, em relação ao custo total e a quantidade de portas reversíveis ternárias dos circuitos sintetizados;
- O impacto que as possíveis orientações (*backward, forward, top-down* e *bottom-up*) no processo de busca têm sobre os circuitos sintetizados;
- Empregar Algoritmos Genéticos ao projeto de circuitos reversíveis ternários como um problema de otimização;

- Mostrar o desempenho de variantes dos Algoritmos Genéticos propostos, como a quantidade de evoluções, o tamanho da população, a seleção por Proximidade dos Inteiros e tamanho dos indivíduos a partir das abordagens Randômica e Menos_Um, as quais também foram desenvolvidas neste trabalho.

2 REVISÃO BIBLIOGRÁFICA

A síntese de circuitos reversíveis, binários ou ternários, é definida pela habilidade de gerar um circuito eficiente a partir de uma determinada especificação reversível de tamanho arbitrário. É importante ressaltar que existem duas alternativas para início do procedimento de síntese. A primeira é a partir de uma especificação reversível, e a segunda de uma especificação em que as restrições de reversibilidade não são cumpridas, ou seja, uma função ternária não reversível. Dessa forma, antes de realizar a síntese, adiciona-se a especificação ternária inicial, variáveis de entrada extras, cujo objetivo é tornar a especificação não reversível em reversível. Tais variáveis extras são denominadas, na literatura especializada, de *ancillary lines* ou *constant inputs* (do inglês: linhas auxiliares/entradas constantes).

Na concepção de circuitos reversíveis, as entradas constantes são representadas por meio da adição de uma linha extra ao circuito, que é composto por m linhas (m: número de variáveis de entrada e de saída). Quando adicionadas ao circuito, as linhas representam uma unidade de informação, aumentando o custo e a complexidade de implementação. Poucos trabalhos têm sido propostos para a síntese de circuitos reversíveis ternários em relação às inúmeras contribuições encontradas no domínio binário, principalmente os métodos que não se utilizam de entradas constantes e/ou linhas auxiliares (YANG et al., 2005; MILLER; MASLOV; DUECK, 2006; KOLE et al., 2017). Outros métodos conhecidos empregam o uso de linhas auxiliares, pois o procedimento de síntese é iniciado a partir de uma especificação ternária não reversível (MANDAL; CHAKRABARTI; SUR-KOLAY, 2011; LI et al., 2013; RANI, 2016; BASU et al., 2016).

Li et al. (2013) propuseram a síntese de circuitos ternários não reversíveis, utilizando a biblioteca de portas ternárias Swap, Not e Toffoli (SNT) para a realização dos circuitos. Este trabalho leva em consideração a combinação da lógica dos circuitos convencionais e dos quânticos. Com a biblioteca de portas SNT utilizada no procedimento de síntese, o método apresenta como transformar especificações ternárias não reversíveis em reversíveis. Combinando as notações da álgebra abstrata, em específico a teoria de grupo, foi possível sintetizar os circuitos não reversíveis, adicionando $(m + [\log_3(\mu)] - n)$ entradas constantes, em que m representa o número de saídas originais da função e n representa o número de entradas originais da função. E, $[\log_3(\mu)]$ de saídas *garbages*, em que μ é a quantidade máxima de vezes que a saída se repete na especificação não reversível. De acordo com os resultados principais, pôde ser visto que os circuitos quânticos ternários diferem dos circuitos quânticos binários e

que o número de entradas constantes e saídas *garbages* adicionadas para tornar as funções reversíveis foram mínimas quando comparados com os métodos de Babu et al. (2004) e Mandal, Chakrabarti e Sur-kolay (2011). Para o exemplo apresentado, de um somador completo, o método de Babu et al. (2004) gerou 2 entradas constantes e 3 saídas *garbages*, enquanto que a proposta de Li et al. (2013) gerou 1 entrada constante e 2 saídas *garbages*. Todavia, no trabalho de Babu et al. (2004), foi definida uma porta ternária NG que, apesar de acrescer o custo do circuito, gerando entradas constantes e saídas *garbages*, requereu uma quantidade mínima de portas para a realização do circuito completo, ao passo que a síntese proposta por Li et al. (2013) gerou um circuito com mais de 150 portas reversíveis ternárias.

Com os exemplos apresentados, pode-se concluir que o método de Li et al. (2013) utiliza 50% menos variáveis para o meio somador e 66,7% para o somador completo. No entanto, o número de portas utilizadas é 30,5 vezes maior para o somador completo e 20 vezes para o meio somador em comparação à proposta de Mandal et al. (2011).

Rani et al. (2017) também desenvolveram um método construtivo para a síntese reversível ternária com base na teoria de grupo, em que são requeridas as portas reversíveis ternárias Toffoli, Toffoli^{+} e NOT (T_T, T_{T^+} e NT) para realizar a síntese. Funções booleanas são utilizadas como especificações de entrada do método, representadas na forma de permutações. Estas permutações são, então, decompostas em ciclos de permutação de tamanho 3. Posteriormente, um mapeamento das portas T_T, T_{T^+} e NT sobre cada ciclo é realizado, em que a união das portas obtidas com os ciclos formam o circuito reversível ternário final. Os resultados experimentais apresentados mostraram que o método construtivo proposto, ao implementar um meio-somador, gera um circuito com menor número de linhas auxiliares e saídas *garbages*, quando comparado ao método proposto por Mandal et al. (2011). Além disso, apresenta uma melhora de 27% sobre o trabalho de Li et al. (2013). Contudo, a restrição do método a portas reversíveis ternárias com o máximo 3 variáveis de entrada acresce demasiadamente a quantidade de portas necessárias para sintetizar o mesmo *benchmark* de Li et al. (2013) e Mandal et al. (2011).

Divergindo dos trabalhos mencionados, Miller, Maslov e Dueck (2006) apresentaram um algoritmo heurístico que explora a síntese de circuitos reversíveis ternários livres de linhas auxiliares, utilizando-se do conjunto de portas reversíveis MLV propostas por De Vos et al (2002). O método de síntese apresentado é uma extensão do método binário MMD, desenvolvido pelos mesmos autores em 2003 (MILLER; MASLOV; DUECK, 2003).

Basicamente, o procedimento de síntese proposto pelos autores consiste em encontrar uma sequência de permutações elementares para realizar uma função reversível sem linhas

auxiliares, em que as funções de entrada do algoritmo são ordenadas lexicograficamente. Cabe ressaltar que as funções sintetizadas pelo método devem incialmente ser reversíveis, uma vez que não é realizada nenhuma operação para transformar funções não reversíveis em reversíveis. Além disso, o método permite que os circuitos sejam sintetizados em duas direções simultaneamente, *backward* e *forward*. Na direção *backward*, para uma dada especificação reversível, inicia-se a síntese a partir da função de saída para a função de entrada, e na direção *forward*, o procedimento inverso é aplicado. Esta possibilidade de execução bidirecional possui a vantagem de realizar circuitos mais simples e, portanto, menos dispendiosos.

Experimentos exaustivos foram realizados para todas as possíveis funções reversíveis ternárias com duas variáveis. Como cada função contêm 9 vetores de entrada (00, 01, 02, 10, 11, 12, 20, 21 e 22), estes vetores podem ser permutados em 362880 ordens distintas, ou seja, 9!. Os resultados mostraram que, embora o algoritmo não gere uma solução mínima garantida, sempre encontra uma solução e estas estão próximas do ponto mínimo. Este trabalho, dentre os conhecidos na literatura especializada, é o único que realiza testes exaustivos possibilitando comparações concretas e conclusivas em relação às alternativas de síntese propostas.

Além do trabalho de Miller, Maslov e Dueck (2006), Yang et al. (2006) propuseram uma investigação com a finalidade de encontrar portas reversíveis ternárias universais que possam ser empregadas na síntese de circuitos reversíveis ternários também sem linhas auxiliares. Foi demonstrado que as portas reversíveis ternárias Swap, NOT e Toffoli são universais para realização de qualquer circuito reversível ternário de tamanho arbitrário $n \times n$, em que n é a quantidade de variáveis/bits de entrada e saída. Além disso, foi desenvolvido um procedimento construtivo que utiliza tais portas lógicas reversíveis ternárias para projetar circuitos. Este procedimento construtivo compreende a realização de uma função reversível ternária na forma de ciclos de permutação. Os autores mostraram que a teoria de grupo pode ser aplicada no domínio ternário, assim como no binário, e que este procedimento é convergente. Contudo, são apresentados somente dois exemplos da síntese realizada sobre ciclos de permutação, utilizando-se das portas ternárias Swap, NOT e Toffoli. Os custos dos circuitos reversíveis ternários não são mencionados, os quais poderiam ser em relação ao número de portas ou custo quântico.

Em relação à síntese de circuitos reversíveis/quânticos ternários empregando-se algoritmos genéticos, a primeira abordagem foi proposta por Khan e Perkowski (2004). O método pôde ser aplicado tanto para funções reversíveis ternárias completamente especificadas quanto para incompletamente especificadas, e as portas GTG (Portas Toffoli Generalizadas) foram escolhidas no procedimento de síntese. Foi apresentado no trabalho o exemplo de um

circuito reversível ternário de um meio-somador com 8 portas GTG. Os autores mencionaram que outros circuitos gerados não foram apresentados devido à impossibilidade de comparações.

Utilizando-se de outro conjunto de portas reversíveis ternárias, Khanoma, Kamalb e Khana (2008) também propuseram um algoritmo genético para a síntese de circuitos reversíveis/quânticos ternários. As portas adotadas foram do tipo MS (MUTHUKRISHNAN; STROUD, 2000). Para o exemplo experimental apresentado, obteve-se um circuito reversível ternário de um meio-somador, com custo quântico igual a 19, isto é, o circuito compõe-se de 19 portas reversíveis ternárias MS. Uma vez que os circuitos gerados podem conter portas redundantes, também foi proposta uma abordagem de otimização pós-síntese, em que estas portas são substituídas com a finalidade de reduzir o custo quântico. Foram apresentadas três alternativas para as possíveis substituições. Uma destas foi aplicada para o circuito gerado por meio da síntese proposta, no qual duas portas controladas em cascata no circuito foram substituídas por uma única porta, também controlada. Destaca-se que os controles devem ser iguais para que a substituição seja feita. Assim, após a otimização, duas portas reversíveis ternárias +1 com controles iguais 2 foram substituídas por uma única porta +2 de controle 2 e, portanto, o custo do circuito foi reduzido de 19 para 18.

Analogamente, no trabalho de Deibuk e Biloshytskyi (2015), foi desenvolvido um algoritmo genético clássico para a síntese reversível ternária, empregando-se portas reversíveis ternárias MS. O método proposto para a codificação dos cromossomos e a escolha adequada dos parâmetros do algoritmo permitem a obtenção de circuitos melhores em relação aos comparadores reversíveis ternários de 1-qutrit e *n*-qutrits, apresentados por Khan (2008) e Zadeh e Haghparast (2011). Os qutris, no domínio quântico, são utilizados para referir cada unidade de informação ternária, analogamente aos bits, referidos na computação convencional.

Os parâmetros utilizados para mensurar as melhorias dos circuitos obtidos foram o tempo de atraso, o custo quântico, e o número de linhas auxiliares dos circuitos resultantes. A implementação, então, levou à redução do tempo de atraso do dispositivo e do número de linhas auxiliares. Especificamente, reduziu-se o número de linhas auxiliares a 1 e em $2n$-1 para os circuitos dos comparadores com 1-qutrit e *n*-qutrits, respectivamente.

Posteriormente, Hu e Deibuk (2016) propuseram a síntese para dispositivos reversíveis ternários aritméticos com base em algoritmos genéticos adaptativos. Os circuitos foram sintetizados a partir do conjunto completo de portas reversíveis ternárias MS, que podem ser fisicamente implementadas e têm um custo quântico mínimo, segundo os autores. Os circuitos determinados para um meio-somador, um somador-completo, um meio-subtrator e um completo-subtrator têm melhores resultados com relação ao custo quântico, entradas *constants*,

saídas *garbages* e atraso, quando comparados com trabalhos anteriores que tiveram a mesma proposta (KHAN; PERKOWSKI, 2007, ZOBOV; PEKHTEREV, 2009; DEIBUK; BILOSHYTSKYI, 2015; MONFARED; HAGHPARAST, 2016). A vantagem dos circuitos obtidos foi a possibilidade de implementação em tecnologia NMR (Ressonância Magnética Nuclear).

Como pôde ser visto, os trabalhos para a síntese de circuitos reversíveis ternários que recorrem ao uso de algoritmos genéticos têm sido aplicados a propostas comuns e específicas para circuitos aritméticos. Lukac e Perkowski (2010) mostraram um procedimento que difere dos abordados anteriormente. Experimentalmente, a síntese lógica quântica (QLS) heterogênea, usando um Algoritmo Evolutivo com um conjunto de restrições estruturais, foi realizada. Os autores definiram QLS heterogênea como sendo um procedimento para projetar circuitos quânticos, empregando não só portas quânticas com a mesma dimensão, mas combinando portas de diferentes propriedades e dimensões. Nos experimentos, foram utilizadas portas quânticas binárias e ternárias para sintetizar uma determinada função.

Esta abordagem foi benéfica nos *benchmarks* que puderam sintetizar nas condições experimentais definidas. Como tal, o uso de restrições estruturais no Algoritmo Evolutivo gerou uma melhoria promissora no processo de síntese. Além disso, foi demonstrado que o emprego da arquitetura SIMD (*Single Instruction Multiple Data*) para a avaliação de circuitos quânticos permite que o processo seja consideravelmente acelerado. Para comparação, uma busca evolucionária em um circuito quântico de 4-qutrits utilizando uma CPU (*Central Processing Unit*) convencional e, portanto, não dedicada, levou duas a três vezes mais tempo computacional do que quando a matriz do circuito é calculada na GPU *(Graphics Processing Unit*) dedicada. A computação baseada em GPU SIMD é útil para avaliar matrizes muito grandes. Além disso, a separação da avaliação (GPU) e o resto do algoritmo (CPU) resultam na possibilidade de computar circuitos quânticos maiores, além de implementar operadores evolutivos mais complexos sem tempo computacional maior.

A abordagem heterogênea mostrou ser uma abordagem interessante para QLS. E isto, não se deve somente pela incorporação do binário em ternário ou ternário em quaternário, mas sim pela possibilidade de uso de operadores binários em um espaço de lógica ternária. Isso significa que, em sentido estrito, esta abordagem não é uma síntese de lógica quântica de múltiplos valores, mas sim uma combinação de portas quânticas de várias dimensões, (neste caso binário e ternário) para projetar uma função especial.

3 CIRCUITOS REVERSÍVEIS TERNÁRIOS

Neste capítulo são abordadas as propriedades de reversibilidade e suas aplicações nos circuitos reversíveis ternários. Apresentam-se as portas reversíveis ternárias empregadas neste trabalho e por fim as características relevantes para a concepção dos circuitos reversíveis ternários.

3.1 FUNÇÕES REVERSÍVEIS TERNÁRIAS

Definição 1: Uma função totalmente especificada f: $B^n \rightarrow B^m$ é considerada reversível se atender os seguintes critérios (NIELSEN e CHUANG, 2002):

I. A quantidade de entradas e de saídas deve ser igual ($n = m$);
II. Os vetores de entrada e de saída devem ter correspondência bijetiva.

Em que $B \in \{0,1\}$ se a função reversível for binária, e $B \in \{0,1,2\}$ se a função reversível for ternária.

Os dígitos ternários 0, 1 e 2 são referidos como *trits,* analogamente aos *bits* no domínio binário, representando uma unidade de informação ternária.

Uma determinada especificação reversível pode ser representada na forma de tabela de valores (ou tabela verdade). Utiliza-se como exemplo, uma determinada função ternária F = (00, 01, 02, 01, 02, 10, 02, 10, 11) especificada na forma de tabela de valores, Tabela 1. Considera-se que a tabela possui somente duas colunas, em que a primeira é denominada coluna de Entrada, representada pelas variáveis x_{in} e y_{in}, e a segunda é denominada de coluna de Saída, representada pelas variáveis x_{out} e y_{out}. Considera-se também que cada linha da segunda coluna é um vetor de saída, e o mesmo aplica-se para a primeira coluna.

A coluna de entrada contém todas as possíveis atribuições aos vetores de entrada, que são ordenados lexicograficamente, bem como a coluna de saída contém as atribuições dos vetores de saída correspondentes aos de entrada.

Tabela 1 – Especificação de uma função ternária na forma de tabela de valores.

Entrada		**Saída**	
x_{in}	y_{in}	x_{out}	y_{out}
0	0	0	0
0	1	0	1
0	2	0	2
1	0	0	1
1	1	0	2
1	2	1	0
2	0	0	2
2	1	1	0
2	2	1	1

→ Vetor de saída [0 2]

Fonte: Próprio Autor

Analisando a reversibilidade da função ternária especificada na Tabela 1, é possível notar que o primeiro critério de reversibilidade descrito na Definição 1 é atendido, em que a quantidade de variáveis de entradas e de saídas da função são iguais. No entanto, o vetor de saída [0 2] está contido três vezes na coluna de saída, violando a restrição de bijetividade e, portanto, a função não é reversível. Neste caso, para torná-la reversível é necessário eliminar as repetições. Isto é possível adicionando novas variáveis de entrada e/ou saída à função ternária. Analogamente aos circuitos reversíveis binários, a variável de saída extra adicionada a uma função com a finalidade de torná-la reversível é denominada de *garbage* e as variáveis de entrada são denominadas de *entradas constantes* ou *linhas auxiliares*. O termo *constante* provém do fato de que as entradas adicionadas devem ter valores determinados para alcançar a funcionalidade desejada nas saídas.

Para que o número de variáveis de entrada e saída sejam sempre iguais deve-se manter a relação apresentada na Equação 1:

$$\text{entrada} + \text{entrada } \textit{constante/linhas auxiliares} = \text{saída} + \text{saída } \textit{garbage} \qquad (1)$$

O mínimo de saídas *garbages* necessárias para tornar uma função não reversível em reversível é dado por $[\log_3(\mu)]$, em que μ é a quantidade máxima de vezes que um vetor de saída é repetido na tabela de valores correspondente. A quantidade de entradas *constantes,* que devem ser adicionadas, pode ser calculada pela expressão $m + [\log_3(\mu)] - n$, em que m

representa o número de saídas originais da função e *n* representa o número de entradas originais da função (LI et al. 2013).

Na Tabela 1, o vetor de saída [0 2] está contido três vezes nesta tabela. Logo $\mu = 3$ e $\log_3(3) = 1$, e, portanto, no mínimo 1 saída *garbage* deve ser adicionada para eliminar os vetores de saída repetidos. Posteriormente, calcula-se a quantidade de entradas *constantes*. A função possui $m = 2$ saídas originais e $n = 2$ entradas originais. Atribuindo os valores na expressão ($m + [\log_3(\mu)] - n$) tem-se que $2 + [1] - 2 = 1$ entrada *constante.* Deve-se ressaltar que cada uma das *n* variáveis de entrada *constantes* podem assumir independentemente os valores ternários 0, 1 ou 2, assim existirão 3^n possíveis atribuições para as variáveis a serem consideradas na determinação do valor da função. Portanto, $3^3 = 27$ possíveis combinações, como mostra-se na Tabela 2, em que os valores ternários foram atribuídos aleatoriamente, mantendo a correspondência bijetiva, isto é, um único vetor de saída para cada vetor de entrada. Os valores ternários em negrito representam a Tabela 1 inicial (não reversível).

Existe mais do que uma função reversível ternária associada a cada função não reversível, dependendo do número de entradas *constantes* e de *garbages* adicionadas, e das atribuições feitas aos valores ternários. Os circuitos resultantes da sintetização dessas funções reversíveis ternárias incorporadas são diferentes, em relação à quantidade de portas, número de linhas (*trits*) e, portanto, têm custos de implementação diferentes. Consequentemente, o processo de incorporar uma função irreversível em uma função reversível é de importância significativa e permanece um problema aberto ao ser estudado em muitos trabalhos (SASANIAN, 2012).

Tabela 2 – Especificação reversível da função ternária apresentada na Tabela 1.

Entrada			**Saída**		
C_{in}	x_{in}	y_{in}	G_{out}	x_{out}	y_{out}
0	**0**	**0**	0	**0**	**0**
0	**0**	**1**	0	**0**	**1**
0	**0**	**2**	0	**0**	**2**
0	**1**	**0**	1	**0**	**1**
0	**1**	**1**	1	**0**	**2**
0	**1**	**2**	0	**1**	**0**
0	**2**	**0**	2	**0**	**2**
0	**2**	**1**	1	**1**	**0**
0	**2**	**2**	0	**1**	**1**
1	0	0	1	0	0
1	0	0	0	1	2
1	0	1	0	2	2
1	0	2	1	1	1
1	1	0	0	2	1
1	1	1	1	1	2
1	1	2	1	2	0
1	2	0	1	2	1
1	2	1	1	2	2
1	2	2	2	0	0
2	0	0	2	0	1
2	0	1	2	2	2
2	0	2	2	1	0
2	1	0	2	1	1
2	1	1	2	1	2
2	2	0	2	2	0
2	2	1	2	2	1
2	2	2	0	2	0

Fonte: Próprio Autor

Logo, os vetores de saída de uma função reversível define uma permutação dos vetores de entrada, em que para uma determinada especificação reversível ternária com $m = 2$, estes vetores podem ser permutados em 362880 mil ordens distintas, isto é, $3^m!$. Esta quantidade é consideravelmente menor no caso binário com uma quantidade maior de variáveis de entrada $m = 3$, no qual existem $2^m! = 40320$ funções reversíveis binárias. Ressaltando que no caso ternário é 3^m, uma vez que existem 3 valores ternários possíveis 0, 1 e 2.

3.2 PORTAS REVERSÍVEIS TERNÁRIAS

Uma primeira decisão básica para realização da síntese de circuitos reversíveis ternários é a escolha das portas reversíveis ternárias. Durante a síntese, uma especificação reversível é implementada utilizando-se de um conjunto básico de portas.

Definição 2: Uma porta lógica é considerada reversível se contiver a mesma quantidade de entradas e saídas, e realizar uma função bijetiva (TOFFOLI, 1980).

Introduz-se um conjunto de cinco portas reversíveis ternárias empregados na execução deste trabalho: $\oplus 1$, $\oplus 2$, $\mathbf{P}_{12}$, $\mathbf{P}_{01}$ e NOT. Na Tabela 3, encontram-se as operações elementares realizadas por cada porta reversível ternária, bem com os símbolos que serão representados na ilustração dos circuitos reversíveis ternários.

Tabela 3 - Portas reversíveis ternárias, respectivas operações elementares e símbolos.

Notação	**Operação Elementar**	**Representação**
NOT	$\mathbf{N}(x) = 2x \oplus 2$	N
$\mathbf{P}_{01}$	$\mathbf{P}_{01}(x) = 2x \oplus 1$	01
$\mathbf{P}_{12}$	$\mathbf{P}_{12}(x) = 2x$	12
$\oplus 1$	$x \oplus 1$	+1
$\oplus 2$	$x \oplus 2$	+2

Fonte: Próprio Autor

As operações elementares realizadas pelas portas reversíveis ternárias são especificadas no Campo de Galoá 3 (GF_3) (KLIMOV et al., 2004). O GF_3 consiste em um conjunto de três elementos T = {0, 1, 2} e em duas operações básicas que podem ser realizadas sobre tais elementos. Estas operações são de multiplicação *módulo 3* e adição *módulo 3* (denotada por $\oplus$), que estão exemplificadas na Tabela 4. O operador de adição *módulo 3* usa o mesmo símbolo da operação XOR ($\bigoplus$) no domínio binário, pois, matematicamente, a operação XOR binária pertence ao Campo de Galoá 2 (*módulo 2*).

Tabela 4 - Operações de adição e multiplicação *módulo 3* referentes ao Campo de Galoá 3.

$\oplus$	**0**	**1**	**2**
0	0	1	2
1	1	2	0
2	2	0	1

(a)

$\bullet$	**0**	**1**	**2**
0	0	0	0
1	0	1	2
2	0	2	1

(b)

Fonte: Próprio Autor.

Na Tabela 5, apresentam-se as operações elementares de cada porta reversível ternária sob uma única variável de entrada *x*. Sabe-se, então, por exemplo, que se $x = 0$ e a porta reversível ternária NOT é aplicada sobre *x*, o valor resultante é de 2. Denominam-se tais portas, com apenas uma variável de entrada, de portas reversíveis ternárias simples, em que as operações nos circuitos são sempre realizadas independentemente.

Tabela 5 - Operação de cada porta reversível ternária sob *x*.

Variável	**Operações sobre a variável *x***				
x	$NOT_{(x)}$	$\mathbf{P}_{01(x)}$	$\mathbf{P}_{12(x)}$	$x \oplus 1$	$x \oplus 2$
0	2	1	0	1	2
1	1	0	2	2	0
2	0	2	1	0	1

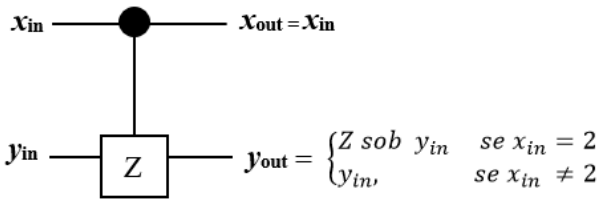
Fonte: Próprio Autor

Todas as portas reversíveis ternárias utilizadas, quando controladas, serão do tipo *Muthukrishnan-Stroud gates* (MS) (MUTHUKRISHNAN; STROUD, 2000), isto é, as portas serão ativadas no circuito se a variável de controle estiver no estado 2. Senão, permanecem inalteradas e comportando-se com uma função identidade.

Definição 3: Uma porta reversível ternária G (x_{in}; y_{in}), do tipo MS, define que o valor de entrada x_{in} controla o valor de saída y_{out}. Realiza-se a operação *Z*, se x_{in} for igual 2. Sendo que $Z \in \{\oplus 1, \oplus 2, \mathbf{P}_{12}, \mathbf{P}_{01}, NOT\}$. Se $x_{in} \neq 2$, então, $y_{out} = y_{in}$.

A representação gráfica de uma porta reversível ternária do tipo MS é ilustrada na Figura 1, em que um círculo preto preenchido é usado para representar o controle 2.

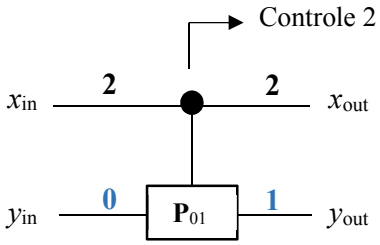
Figura 1 - Porta reversível ternária do tipo MS.



Fonte: Adaptada de Muthukrishnan e Stroud (2000).

Na Figura 2, ilustra-se o comportamento da porta reversível ternária $\mathbf{P}_{01}$. As variáveis de entrada são x_{in} e y_{in}, e as respectivas variáveis de saída são x_{out} e y_{out}. Neste caso, x_{in} é a variável de controle e uma vez que $x_{in} = 2$ a porta $\mathbf{P}_{01}$ é ativada e, portanto, realiza-se a operação elementar $2x \oplus 1$ referente à porta em y_{in}.

Figura 2- Porta reversível ternária $\mathbf{P}_{01}$.

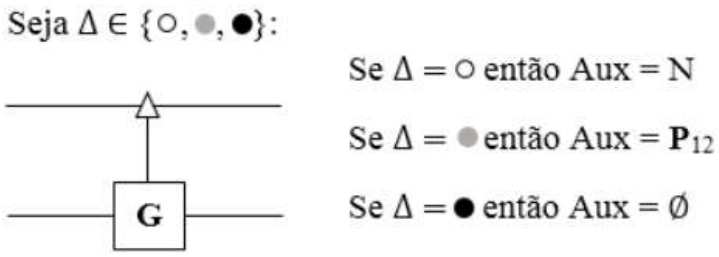


Fonte: Próprio Autor.

Conforme mencionado, as portas reversíveis ternárias controladas do tipo MS são ativadas somente quando a variável de controle consiste no estado/valor 2. Neste caso, um ponto preto é utilizado para simbolizar tal controle. Analogamente ao caso binário, introduz-se uma representação simbólica para designar as portas reversíveis ternárias quando controladas por 0 ou 1. São utilizados pontos brancos e cinzas para representar os controles 0 e 1, respectivamente. Contudo, uma vez que estes controles são representados simbolicamente, faz-se necessária a adição de portas simples consideradas auxiliares nas respectivas implementações. As portas simples auxiliares, cumprem a tarefa de transformar os controles 0 e 1 em 2, para que as portas possam ser ativadas tanto pelo controle 2 quanto pelos controles 0

e 1. Esta situação é ilustrada na Figura 3 por meio de um modelo generalizado para portas reversíveis ternárias do tipo MS. Comumente, as portas simples auxiliares não são mostradas no diagrama do circuito.

Figura 3 - Modelo generalizado para portas reversíveis ternárias do tipo MS.



Fonte: Próprio Autor

As portas reversíveis são operadores lineares e, por conseguinte, podem ser representadas por matrizes unitárias. Na computação quântica, as portas reversíveis devem ser definidas como matrizes unitárias, uma vez que as representações matriciais são necessárias para calcular a função de transferência. A função de transferência tem por objetivo verificar o comportamento correto dos circuitos. Os detalhes da função serão abordados na seção 3.4.

No caso da computação reversível, quando as portas são matrizes de valor inteiro, o requisito unitário reduz-se à ortogonalidade.

Para a porta reversível ternária simples NOT, isto é, sem controle, tem-se a representação matricial $\begin{bmatrix}0 & 0 & 1\\0 & 1 & 0\\1 & 0 & 0\end{bmatrix}$ e quando aplicada aos valores ternários 0,1 e 2 os seguintes resultados são obtidos:

$$\begin{bmatrix}0 & 0 & 1\\0 & 1 & 0\\1 & 0 & 0\end{bmatrix}\cdot\begin{bmatrix}\mathbf{0}\\\mathbf{1}\\\mathbf{2}\end{bmatrix}=\begin{bmatrix}2\\1\\0\end{bmatrix} \tag{2}$$

Como pode-se notar na Equação 2, a matriz de permutação da porta reversível ternária NOT é simétrica e autoinversa. Similarmente, as matrizes que correspondem às portas reversíveis ternárias $\mathbf{P}_{01}$ e $\mathbf{P}_{12}$ são simétricas e autoinversas. Esta situação é mostrada na Equação 3. Logo, as matrizes de permutação das portas NOT, $\mathbf{P}_{01}$, P_{12} são ortogonais.

$$\mathbf{P}_{01}=\begin{bmatrix}0 & 1 & 0\\1 & 0 & 0\\0 & 0 & 1\end{bmatrix}\cdot\begin{bmatrix}\mathbf{0}\\\mathbf{1}\\\mathbf{2}\end{bmatrix}=\begin{bmatrix}1\\0\\2\end{bmatrix};\qquad \mathbf{P}_{12}=\begin{bmatrix}1 & 0 & 0\\0 & 0 & 1\\0 & 1 & 0\end{bmatrix}\cdot\begin{bmatrix}\mathbf{0}\\\mathbf{1}\\\mathbf{2}\end{bmatrix}=\begin{bmatrix}0\\2\\1\end{bmatrix} \tag{3}$$

Em particular, as matrizes referentes às portas $\oplus 1$ e $\oplus 2$, requerem uma análise especial. Estas portas são baseadas na matriz X-Pauli (PAULI, 1933), em que a operação elementar corresponde à adição de 2 *módulo 3*, como pode ser visualizado na Equação 4.

$$\text{X- Pauli}_{(y)} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \\ \mathbf{2} \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix} = y \oplus 2 \tag{4}$$

A matriz X-Pauli não é autoinversa. Contudo, X-Pauli é ortogonal e sua inversa é equivalente à sua matriz X-Pauli transposta, X^T. Uma vez que a implementação de X-Pauli consiste em adicionar 2 *módulo 3* ao argumento, e X^T é sua inversa. A operação elementar de X^T consiste em adicionar 1 *módulo 3*. Esta situação é apresentada na Equação 5

$$\text{X- Pauli}^{T}_{(y)} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \\ \mathbf{2} \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix} = y \oplus 1 \tag{5}$$

Definidas as matrizes unitárias referentes às portas reversíveis ternárias simples, definem-se as matrizes que representam as portas controladas por 0, 1 e 2.

O tamanho da matriz unitária que descreve uma porta reversível ternária é $3^m \times 3^m$, em que *m* corresponde à quantidade de variáveis de entrada/saída. Uma vez que neste trabalho todas as especificações reversíveis possuem duas variáveis, as matrizes unitárias que definem as portas com controles apresentam a dimensão de $3^2 \times 3^2$.

Dada uma matriz quadrada 9×9, define-se que Diag(**A, B, C**) denota as matrizes 3×3 da diagonal principal, cujos os elementos restantes são iguais a 0. Define-se também ***M*** como sendo a matriz unitária que representa umas das portas reversíveis ternárias simples, que foram apresentadas nas Equações 1, 2, 3 e 4. Bem como C***M*** denota uma realização controlada de ***M***. Isto posto, se ***M*** é controlada por 0. Então, a diagonal principal da matriz unitária de C***M*** = Diag(***M*****, I, I**). Se ***M*** é controlada por 1, C***M*** = Diag**(I, *M*, I**), e se ***M*** é controlada por 2, C***M*** **=** Diag**(I, I, *M*),** em que **I** é uma matriz identidade. Por meio da Figura 4 tais definições são ilustradas.

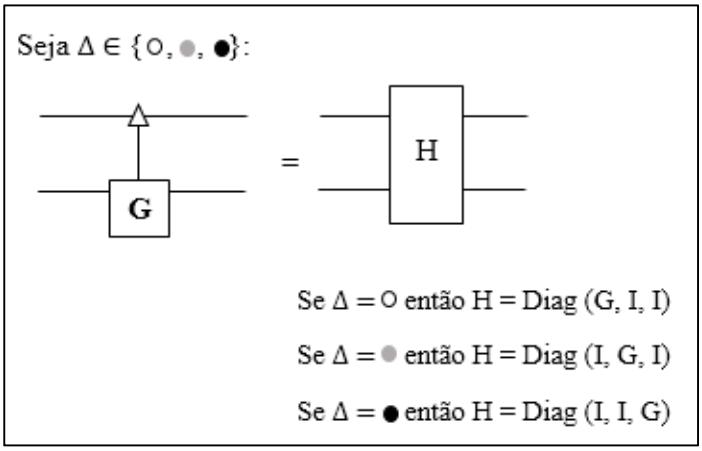
Ademais, se $\boldsymbol{M_1}$ e $\boldsymbol{M_2}$ são matrizes de portas dispostas simultaneamente como portas simples nas linhas superior e inferior de um circuito de 2 entradas, respectivamente, sua representação matricial conjunta é obtida por meio do produto de Kronecker $\boldsymbol{M}_1 \otimes \boldsymbol{M}_2$ (NIELSEN e CHUANG, 2002). E, se uma única porta simples $\boldsymbol{M}_1$ é disposta paralelamente no

circuito, a sua representação matricial será $\boldsymbol{M}_1 \otimes \mathbf{I}$, se a porta se encontra na primeira linha, ou $\mathbf{I} \otimes \boldsymbol{M}_1$ se a porta está disposta na segunda linha do circuito.

Definição 4: Assumindo $A \in \mathbb{F}^{n \times m}$ e $B \in \mathbb{F}^{l \times k}$. Assim, o produto de *Kronecker* $A \otimes B \in \mathbb{F}^{n\,l \times m\,k}$ de A e B é a matriz particionada, como mostra-se na Equação 6:

$$A \otimes B \triangleq \begin{bmatrix} A_{(1,1)}B & A_{(1,2)}B & \cdots & A_{(1,m)}B \\ \vdots & & & \vdots \\ A_{(n,1)}B & A_{(n,2)}B & \cdots & A_{(n,m)}B \end{bmatrix} \tag{6}$$

Figura 4 – Modelo para a representação de portas reversíveis ternárias controladas na forma de matriz unitária.



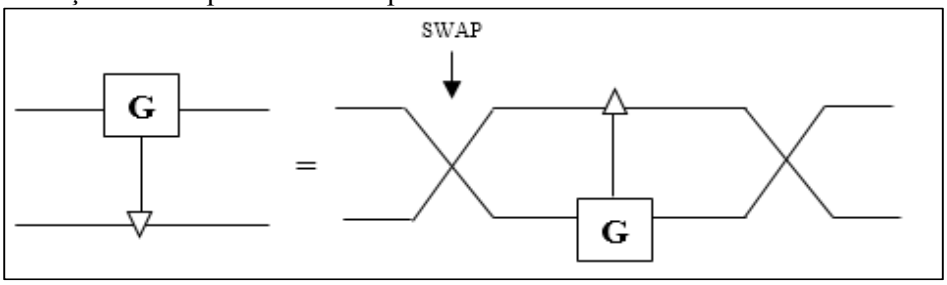
Fonte: Próprio Autor

Define-se que $\boldsymbol{G}\langle v_z \rangle$ denota uma determinada porta reversível ternária $\boldsymbol{G}$ da Tabela 3, controlada pelo trit $v \in \{0, 1, 2\}$, cujo trit de controle encontra-se na linha $z \in \{x, y\}$. Como exemplo, apresenta-se na Equação 7 a matriz correspondente a porta reversível ternária $\mathbf{P}_{12}\langle 2_x \rangle$. Pode-se notar que a primeira e a segunda matriz 3 × 3 da diagonal principal são matrizes identidade e a última matriz, em negrito, refere-se à porta $\mathbf{P}_{12}$ simples.

$$\text{Diag}(\mathbf{I}, \mathbf{I}, \mathbf{P}_{12}) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{0} & \mathbf{0} & \mathbf{1} \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{0} & \mathbf{1} & \mathbf{0} \end{bmatrix} \quad (7)$$

Como exemplificado na Figura 4, o controle da porta reversível ternária ***M*** encontra-se na primeira linha *x*. Entretanto, os controles também podem ser realizados na linha *y*. Para representação matricial das portas com controle na linha *y*, faz-se necessário a utilização da porta reversível SWAP, que tem por finalidade realizar o cruzamento entre duas linhas do circuito. A porta SWAP é a realização da porta Feynman sem controle (FEYNMAN, 1986). A situação descrita é apresentada na Figura 5.

Figura 5 – Ilustração do comportamento da porta reversível SWAP.



Fonte: Próprio Autor

A matriz unitária de uma porta reversível SWAP com $m = 2$ é descrita na Equação 8. Define-se então que a matriz unitária de uma porta reversível ternária com controle na linha *y* corresponde a $\text{SWAP} * \text{Diag}(\mathbf{A}, \mathbf{B}, \mathbf{C}) * \text{SWAP}$.

$$\text{SWAP} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{8}$$

Considerando como exemplo a porta $\mathbf{P}_{12}\langle 2_y \rangle$, apresenta-se na Equação 9 a matriz unitária correspondente.

SWAP * Diag **(I, I, P**$_{12}$**)** * SWAP =

$$= \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{\text{SWAP}} * \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{0} & \mathbf{0} & \mathbf{1} \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{0} & \mathbf{1} & \mathbf{0} \end{bmatrix}}_{\text{Diag}(\mathbf{I, I, P}_{12})} * \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{\text{SWAP}}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{9}$$

O conjunto de portas reversíveis ternárias empregadas durante procedimentos de síntese deve ser completo, isto é, deve ser capaz de realizar todas as especificações reversíveis empregando somente as portas que o contêm. É simples de comprovar que o conjunto de portas {**I**, $\mathbf{P}_{12}$, $\mathbf{P}_{01}$,

NOT} é um conjunto completo, uma vez que, as operações elementares referentes a estas quatro portas são suficientes para gerar as outras portas ⊕1 e ⊕ 2. Além disso, o conjunto {**I**, NOT, $\mathbf{P}_{01}$} também é completo, pois $\mathbf{P}_{12}$ = NOT · $\mathbf{P}_{01}$ · NOT e, por fim, o conjunto {NOT**,** $\mathbf{P}_{01}$} é o menor conjunto completo, pois ambas as portas NOT e $\mathbf{P}_{01}$ são autoinversas e com isso geram todas as outras portas. A interdependência destas portas é mostrada na Tabela 6, em que **I** corresponde a matriz identidade.

Tabela 6 - A interdependência das portas reversíveis ternárias {⊕1, ⊕2, $\mathbf{P}_{12}$, $\mathbf{P}_{01}$, NOT, I}.

→	**I**	**NOT**	$\mathbf{P}_{01}$	$\mathbf{P}_{12}$	**⊕2**	**⊕1**
I	I	NOT	P01	P12	⊕2	⊕1
NOT	NOT	I	⊕2	⊕1	P01	P12
$\mathbf{P}_{01}$	P01	⊕1	I	⊕2	P12	NOT
$\mathbf{P}_{12}$	P12	⊕2	⊕1	I	NOT	P01
⊕2	⊕2	P12	NOT	P01	⊕1	I
⊕1	⊕1	P01	P12	NOT	I	⊕2

Fonte: Próprio Autor

Pode-se também concluir que o conjunto de portas, incluindo a identidade, constitui um grupo multiplicativo não abeliano e, portanto, está completo. Os conjuntos podem ter uma pequena cardinalidade, bem como pode incluir portas reversíveis adicionais. Portas adicionais são redundantes, contudo permitem que o algoritmo seja mais flexível e realize circuitos com menor número de portas reversíveis ternárias em comparação com os circuitos realizados com um conjunto completo mínimo, como proposto por Miller, Maslov e Dueck (2006).

3.3 CIRCUITOS REVERSÍVEIS TERNÁRIOS

Circuitos reversíveis, binários ou ternários, podem ser implementados em diferentes tecnologias convencionais e quânticas. A implementação de circuitos reversíveis na tecnologia CMOS foi estudada em De Vos (2010) e em De Vos (2012). No entanto, para obter vantagens sob o uso das propriedades de reversibilidade, como por exemplo, menor dissipação de calor,

os circuitos reversíveis precisam ser empregados em infraestruturas reversíveis, como tecnologias quânticas.

Existem algumas propriedades válidas para circuitos convencionais que não estão presentes nos circuitos reversíveis:

- Não são permitidos *loops,* ou seja, *feedback* de uma parte do circuito para outra. Logo, circuitos reversíveis são acíclicos;
- *Fan-out* entre as portas reversíveis também não são permitidos.

Definição 5: Um circuito ternário é considerado reversível se for realizado por uma cascata acíclica de portas reversíveis ternárias livres de *fan-out*, cujas portas reversíveis ternárias realizam funções bijetivas.

Consequentemente, um circuito reversível ternário também realiza uma bijeção, isto é, todo vetor com as atribuições dos valores de entrada deve corresponder a um único vetor com as atribuições de saída.

A partir da Definição 5, sabe-se que se apenas os vetores de saída de um circuito reversível são conhecidos, os vetores de entrada podem ser calculados sem ambiguidade. Dessa forma, torna-se evidente que se um circuito é reversível tanto os vetores de entrada quanto os de saída têm a mesma dimensão. A notação $m \times m$ é comumente utilizada para designar um circuito reversível com m entradas e m saídas.

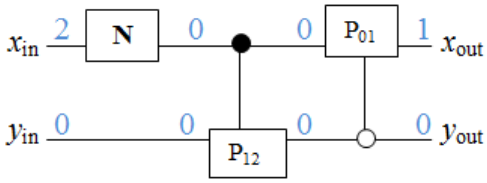
O comportamento dos circuitos reversíveis é diferente dos circuitos digitais convencionais. As informações são processadas na forma de uma operação unitária, que transporta os valores ternários de entrada para à saída ou vice-versa.

Ilustra-se, por meio da Figura 6, um circuito reversível ternário 2×2 que realiza a função reversível ternária $f = (20, 22, 21, 00, 11, 12, 10, 01, 02)$. O circuito é composto por uma cascata de três portas reversíveis ternárias, em que a primeira porta NOT é simples, pois não tem controle. A porta intermediária $\mathbf{P}_{12}$ é controlada por 2, e por último a porta $\mathbf{P}_{01}$ é controlada por 0. Ressalta-se que os trits de saída da primeira porta são as entradas para a segunda, e assim sucessivamente. A saída final é obtida a partir dos trits de saída da última porta.

Atribuindo como valores de entrada $x_{in} = 2$ e $y_{in} = 0$, após a operação $2x \oplus 2$ referente à primeira porta reversível ternária NOT, x_{in} é permutado para 0, pois $2x \oplus 2 = 0$. A segunda variável de entrada $y_{in} = 0$ permanece inalterada, como ilustra-se na Figura 6. Os valores de saída $x_{in} = 0$ e $y_{in} = 0$ resultante da operação da primeira porta tornam-se entradas para a próxima

porta $\mathbf{P}_{12}$. Neste caso, a porta $\mathbf{P}_{12}$ controlada por 2 não é ativada, uma vez que o trit de controle é igual a 0. Assim, os valores permanecem inalterados, tornando-se entradas para a última porta $\mathbf{P}_{01}$. Logo, a porta $\mathbf{P}_{01}$ é ativada, pois é controlada por 0 e o trit de entrada y_{in} também é 0. A operação elementar correspondente $2x \oplus 1$ é aplicada sobre 0, resultando em $x_{in} = 1$. Nenhuma operação é realizada sobre o trit de controle. O trit de controle y_{in} permanece igual a 1. Estas situações podem ser visualizadas na Figura 6, em que os valores dos tris estão ilustrados no circuito. Os valores de entradas iniciais eram x_{in}=2 e $y_{\text{in}} = 0$ e resultaram em x_{out} =1 e $y_{out} = 0$. Ressalta-se que como o circuito é reversível o comportamento do circuito pode ser verificado das saídas para as entradas. Se as variáveis de saída x_{out} e y_{out} fossem atribuídas incialmente com os valores 1 e 0, respectivamente, os valores resultantes seriam de 2 e 0 para x_{in}=2 e $y_{\text{in}} = 0$.

Figura 6 - Ilustração do comportamento de um circuito reversível ternário.



Fonte: Próprio Autor

3.3.1 Função de transferência

O tipo de verificação do comportamento do circuito exemplificado na Figura 6 não é usual e, além disso, não é possível verificar manualmente circuitos com dezenas/centenas de portas, garantindo que as saídas obtidas estejam corretas. Funções de transferência são empregadas para que o comportamento do circuito seja calculado a partir de operações de multiplicação sobre as matrizes unitárias referentes às portas reversíveis ternárias. As matrizes unitárias de cada porta foram definidas na seção 3.2 do Capítulo 3.

A saída de uma porta reversível é obtida pela multiplicação da matriz unitária desta porta pelo vetor de dígitos ternários que correspondem a sua entrada. A matriz unitária que representa um circuito reversível/quântico é obtida por meio das matrizes unitárias das portas que o compõe, conforme descrito nas seguintes etapas:

I. A matriz equivalente de duas portas reversíveis ternárias consecutivas $G_1\langle v_z\rangle G_2\langle v_z\rangle$ é o produto das respectivas matrizes $M_1 M_2$.

II. A matriz equivalente de duas portas simples $G_1\langle x\rangle G_2\langle \mathrm{y}\rangle$ dispostas paralelamente no circuito é obtida pelo produto de Kronecker das matrizes correspondentes $M_1 \otimes M_2$.

Considerando o circuito reversível ternário ilustrado na Figura 6, a representação matricial correspondente é obtida por meio do produto das matrizes das portas que o compõe, $NOT\langle x\rangle$, $\mathrm{P}_{12}\langle 2_x\rangle$ e $\mathrm{P}_{01}\langle 0_y\rangle$.

Uma vez que a porta $NOT\langle x\rangle$ é simples e no circuito não se encontra disposta paralelamente com outra porta, então o produto de *Kronecker* deve ser realizado entre a matriz da porta NOT e a matriz identidade de mesma ordem, como descrito na Equação 10.

A matriz da porta $\mathrm{P}_{12}\langle 2_x\rangle = \mathrm{Diag}(\mathbf{I}, \mathbf{I}, \mathbf{P}_{12})$ e da porta $\mathrm{P}_{01}\langle 0_y\rangle = (\mathrm{SWAP} * (\mathrm{Diag}(\mathbf{P}_{01}, \mathbf{I}, \mathbf{I}) * \mathrm{SWAP})$, como definido na seção 3.2. A matriz $\boldsymbol{M}$ resultante do circuito apresentado na Figura 6 é descrita na Equação 11.

$$(\mathrm{NOT} \otimes \mathbf{I}) = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (10)$$

$$\boldsymbol{M} = (\mathrm{NOT} \otimes \mathbf{I}) * \mathrm{Diag}(\mathbf{I}, \mathbf{I}, \mathbf{P}_{12}) * (\mathrm{SWAP} * (\mathrm{Diag}(\mathbf{P}_{01}, \mathbf{I}, \mathbf{I}) * \mathrm{SWAP}) \quad (11)$$

$$\boldsymbol{M} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Obtida a matriz ***M*** que representa o circuito completo é possível verificar o seu comportamento e, além disso, verificar a função reversível ternária que o circuito realiza. A realização de um circuito é obtida por métodos de síntese a partir especificações reversíveis ternárias.

Na Equação 12, realiza-se o produto da matriz resultante ***M*** pelos vetores de entrada ternários. Com os vetores de saída resultantes sabe-se que o circuito da Figura 6 realiza a função reversível ternária f = (20, 22, 21, 00, 11, 12, 10, 01, 02).

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 2 \\ 1 & 0 \\ 1 & 1 \\ 1 & 2 \\ 2 & 0 \\ 2 & 1 \\ 2 & 2 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 2 & 2 \\ 2 & 1 \\ 0 & 0 \\ 1 & 1 \\ 1 & 2 \\ 1 & 0 \\ 0 & 1 \\ 0 & 2 \end{bmatrix} \quad (12)$$

3.3.2 Métricas de avaliação de desempenho

Tal como para os circuitos lógicos tradicionais, a complexidade de circuitos reversíveis é medida por meio de diferentes métricas. É comum em projetos de circuitos reversíveis mensurar a sua complexidade em termos de um custo quântico teórico, seguindo a convenção introduzida em Barenco et al. (1995), visto que tecnologias quânticas são consideradas as principais áreas de aplicação dos circuitos reversíveis, binários ou ternários. Além disso, para diferentes tecnologias quânticas, o custo concreto de uma determinada porta reversível ou

quântica pode ser diferente, associados à dificuldade de implementação e das características da tecnologia.

Diante disso, nota-se que as métricas de custos diferem significativamente, dependendo do modelo de custo aplicado. O número de portas em cascata é uma medida simples, contudo é a métrica mais independente da tecnologia que está sendo aplicada. Assim, a contagem de porta é muitas vezes usada para avaliar a qualidade de um circuito reversível.

Para comparar a complexidade das realizações, duas métricas são propostas. A quantidade de portas simples e controladas, independentemente do valor de controle, e o custo total, no qual portas simples são atribuídas a um custo igual a 1, controladas por 2 portas a um custo igual a 2 e portas controladas por 0 ou 1 atribuídas a um custo igual a 4. Assim, O custo total de um circuito reversível ternário é definido pelo somatório do custo individual de cada porta reversível ternária. As portas controladas por 0 ou 1 possuem maior custo por causa das duas portas simples auxiliares necessárias na realização do circuito.

Com a finalidade de simplificar a representação do circuito reversível ternário, as portas simples auxiliares necessárias para implementar os controles 0 e 1 não serão ilustradas.

4 ALGORITMOS DE SÍNTESE DE CIRCUITOS REVERSÍVEIS TERNÁRIOS MMD E MMD *plus*

4.1 ALGORITMO DE TRANSFORMAÇÃO MMD

O algoritmo de transformação MMD (acrônimo das iniciais dos nomes dos autores), proposto por Miller, Maslov e Dueck (2003), foi desenvolvido para a síntese de circuitos reversíveis binários sem a adição de linhas auxiliares. Posteriormente, os autores mostraram que o algoritmo MMD poderia ser estendido para a síntese de circuitos reversíveis ternários (MILLER; MASLOV; DUECK, 2006), o qual é denominado neste trabalho de MMD_T.

A metodologia originária do algoritmo MMD consiste em encontrar uma sequência de permutações para transformar uma determinada função reversível em uma função identidade. Em cada permutação é obtida uma porta reversível que a realiza. O conjunto de tais portas forma uma cascata de portas. Esta cascata compreende o circuito resultante, que deve ser realizado da saída para a entrada.

No procedimento iterativo de síntese do algoritmo MMD, a especificação reversível de entrada deve ser representada na forma de tabela de valores (ou tabela verdade), em que a primeira coluna contém todos os vetores de entrada, que são ordenados lexicograficamente. Bem como a segunda coluna contêm as possíveis atribuições dos vetores de saída correspondentes aos respectivos vetores de entrada.

Em cada iteração do algoritmo os vetores de saídas são permutados até que se tornem idênticos aos vetores de entrada correspondentes. Isto é, a especificação inicial é permutada em cada iteração até que se seja transformada em uma função identidade. Quando obtém-se uma função identidade o algoritmo é finalizado. Portanto, durante a execução do algoritmo a coluna com os valores de saída torna-se mais similar à coluna com os valores de entrada. A escolha das portas deve ser feita de tal forma que os vetores de saída do topo que já foram modificados, nomeados de prefixo, não podem ser permutados novamente.

4.1.1 Ilustração da principal ideia do algoritmo MMD

Com a finalidade de compactar a descrição das funções reversíveis ternárias, os vetores ternários são comumente representados na forma decimal. Neste sentido, o vetor ternário [0 0] é representado por 0. Sucessivamente, aplica-se esta mesma notação aos demais vetores [0 1] = 1, [0 2] = 2, [1 0] = 3, [1 1] = 4, [1 2] = 5, [2 0] = 6, [2 1] = 7 e por fim [2 2] = 8. Pode-se

também representar a função reversível ternária f_1, mostrada na Tabela 7, como uma permutação em que $f(0) = 6, f(1) = 5, f(2) = 4, f(3) = 2, f(4) = 0, f(5) = 8, f(6) = 3, f(7) = 1$ e $f(8) = 7$.

Tabela 7 – Função reversível ternária f_1 representada na forma de tabela de valores.

Entrada		**Saída**	
x_{in}	y_{in}	x_{out}	y_{out}
0	0	2	0
0	1	1	2
0	2	1	1
1	0	0	2
1	1	0	0
1	2	2	2
2	0	1	0
2	1	0	1
2	2	2	1

Fonte: Próprio Autor

Basicamente, o algoritmo MMD consiste em 5 passos:

I. Dada uma função reversível ternária na forma de tabela de valores, considere cada linha da tabela como sendo *f (i),* em que *i* é inicializado em 0 e vai até a posição 2*m* -1;

II. Se *f (i) = i*, então *i* deve ser incrementado. Senão, aplique as operações elementares referentes às portas reversíveis ternárias estabelecidas até que *f (i) = i*;

a) A porta reversível ternária que produz a menor distância *Hamming* entre o estado atual da função e a função identidade deve ser escolhida.

III. Crie uma nova coluna de saída para representar os valores ternários gerados após a aplicação das operações elementares no passo II;

IV. A partir da última coluna de saída criada, repita os passos II e III até que *f (i) = i* para todo *i*;

V. A sequência de portas resultantes de cada coluna de saída deve ser representada em um circuito reversível. A ordem das portas no circuito deve ser da saída para entrada.

Definição 6: A distância *Hamming d (x, y)* entre dois vetores *x* e *y*, de mesma dimensão, é dada pela quantidade de dígitos em que *x* e *y* diferem.

Para a escolha da operação elementar no subitem II. a), os autores do algoritmo Miller, Maslov e Dueck (2006) definiram que a distância entre a especificação reversível ternária e a função identidade é dada pela Equação 13, em que para dois *m*-vetores de valores ternários *x* e *y* tem-se:

$$d\,(x,y) = \sum_{k=0}^{m-1} |x_k - y_k| \qquad (13)$$

O processo de síntese reversível ternária é ilustrado por meio do algoritmo MMD utilizando-se da função reversível ternária f_1, descrita na Tabela 7. É importante salientar que o processo de busca padrão do algoritmo MMD é considerado *backward* e *top-down*, com as especificações de entrada do algoritmo ordenadas lexicograficamente, como já mencionado. Quando a síntese é iniciada a partir dos vetores de saída, considera-se, então, que o processo de busca é *backward*, ou seja, os vetores de saída são permutados em cada iteração do algoritmo até que tornem-se iguais aos seus vetores de entrada correspondentes. Quando o vetor mais ao topo da tabela de valores é permutado, primeiramente considera-se que é uma busca *top-down* (de cima para baixo). Na primeira coluna da Tabela 7, denominada de Entrada, encontram-se todos os vetores de entrada da especificação reversível ternária. Na segunda coluna encontram-se os vetores de saída que correspondem aos vetores de entrada da primeira coluna. O processo de síntese do MMD é ilustrado abaixo.

1ª Iteração:

Passo I: A variável $f\,(i)$ é inicializada em zero.

Passo II: Analisa-se se $f\,(0) = 0$. É possível notar que o primeiro vetor de saída [2 0] não condiz com o primeiro vetor de entrada correspondente [0 0]. Logo $f\,(0) \neq 0$, e o procedimento de síntese deve ser iniciado (a partir destes vetores). Uma das portas reversíveis ternárias estabelecidas é escolhida para realizar a operação elementar e, dessa forma, fazer com

que o vetor de saída [2 0] permute para [0 0]. Dentre as possibilidades para realizar tal permutação, o algoritmo é orientado para minimizar a distância H*amming* entre o estado atual da função e a função identidade. A porta reversível ternária ⊕1 foi escolhida por produzir a menor distância *Hamming*. Os valores sombreados na coluna de Saída representam a condição de $y_{out} = 0$ e os valores de x_{out} que serão modificados por meio da operação da porta reversível ternária ⊕1. Assim, quando y_{out} for igual a 0, aplica-se ⊕1 sobre o valor de x_{out}, modificando o vetor [2 0] para [0 0].

Passo III: Cria-se uma nova coluna para representar os novos vetores de saída gerados a partir da aplicação da porta ⊕1. O resultado desta operação é apresentado na Tabela 8, cujos vetores de saída sombreados correspondem aos vetores que serão modificados pela porta ⊕1.

Passo IV: Os passos II e III devem ser repetidos até que $f(i) = i$ para todo i.

Passo II: Analisando a última coluna gerada (1), nota-se que $f(0) = 0$ e, portanto, i deve ser incrementado. Analisa-se i novamente. Uma vez que $f(1) = 5$, deve-se aplicar uma das portas reversíveis ternárias para que $f(1) = 1$.

Tabela 8 – Primeira iteração do procedimento de síntese do algoritmo MMD modo *backward top-down*.

Entrada	**Saída**	**1**
x_{in} y_{in}	x_{out} y_{out}	x_{out} y_{out}
0 0	2 0	**0 0**
0 1	1 2	1 2
0 2	1 1	1 1
1 0	0 2	0 2
1 1	0 0	1 0
1 2	2 2	2 2
2 0	1 0	2 0
2 1	0 1	0 1
2 2	2 1	2 1

Se $y = 0$ então $x \oplus 1$

Fonte: Próprio Autor.

Todos os passos do algoritmo são repetidos (iterativamente) até que a última coluna de Saída seja idêntica à coluna de Entrada. Portanto, tem-se uma função identidade e o algoritmo é finalizado. As operações de síntese geradas são ilustradas por meio da Tabela 9. Os vetores de saída em negrito ilustram os prefixos (vetores que já foram modificados), constituindo-se dos valores desejados e, devido às propriedades estabelecidas no algoritmo MMD, não podem ser alterados novamente.

As permutações elementares geradas com o procedimento de síntese levaram a um circuito reversível ternário com 9 portas e com custo total de 28, como apresenta-se na Figura 7. A ordem das portas reversíveis ternárias geradas em cada iteração do algoritmo deve ser representada no circuito da saída para entrada.

Tabela 9 - Procedimento de síntese do algoritmo MMD modo *backward top-down*.

Entrada	Saída	Operações Elementares								
x_{in} y_{in}	x_{out} y_{out}	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0 0	2 0	**0 0**	**0 0**	**0 0**	**0 0**	**0 0**	**0 0**	**0 0**	**0 0**	**0 0**
0 1	1 2	1 2	1 1	**0 1**	**0 1**	**0 1**	**0 1**	**0 1**	**0 1**	**0 1**
0 2	1 1	1 1	1 2	1 2	**0 2**	**0 2**	**0 2**	**0 2**	**0 2**	**0 2**
1 0	0 2	0 2	0 2	0 2	2 2	2 0	**1 0**	**1 0**	**1 0**	**1 0**
1 1	0 0	1 0	1 0	1 0	1 0	1 0	2 0	2 1	**1 1**	**1 1**
1 2	2 2	2 2	2 2	2 2	1 2	1 2	1 2	1 2	1 2	**1 2**
2 0	1 0	2 0	2 0	2 0	2 0	2 2	2 2	2 0	2 0	**2 0**
2 1	0 1	0 1	0 1	2 1	2 1	2 1	2 1	2 2	2 2	**2 1**
2 2	2 1	2 1	2 1	1 1	1 1	1 1	1 1	1 1	2 1	**2 2**

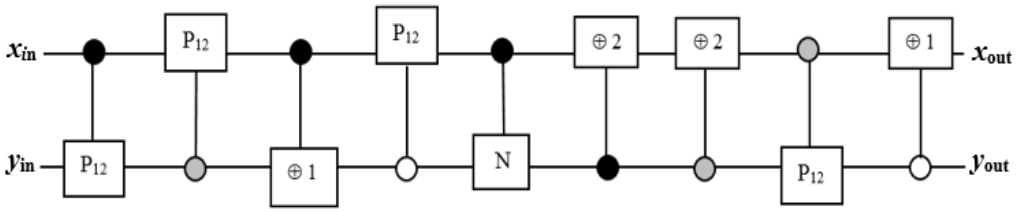
1: Se $y = 0$, então, $x \oplus 1$; **4**: Se $y = 2$, então, $x \oplus 2$; **7**: Se $x = 2$, então, $y \oplus 1$;

2: Se $x = 1$, então, $P_{12}(y)$; **5**: Se $x = 2$, então, N(y); **8**: Se $y = 1$, então, $P_{12}(x)$

3: Se $y = 1$, então, $x \oplus 2$; **6**: Se $y = 0$, então, $P_{12}(x)$; **9**: Se $x = 2$, então, $P_{12}(y)$

Fonte: Próprio Autor.

Figura 7 – Circuito reversível ternário que realiza a função f_1.



Fonte: Próprio Autor

4.2 MMD *plus*

Uma extensão do algoritmo MMD_T à síntese reversível ternária, empregando-se uma específica biblioteca de portas reversíveis ternárias, tem sido proposta e implementada neste trabalho. Denomina-se esta abordagem de MMD *plus*. Cabe ressaltar que durante o procedimento de síntese do MMD_T foram utilizadas as portas reversíveis ternárias propostas por De Vos, Raa e Storme (2002), que são uma adequação das portas reversíveis binárias NOT, Toffoli e Feynman (TOFFOLI, 1980; FEYNMAN,1986). E na abordagem proposta serão utilizadas as portas NOT, $\mathbf{P}_{01,}$ $\mathbf{P}_{12,}$ $\oplus$ 1 e $\oplus$ 2, como mostrado no Capítulo 3.

Uma vez que, em algumas circunstâncias particulares ao processo de síntese, diferentes operações elementares podem ser escolhidas, acarretando em soluções mais dispendiosas ou não, critérios de custos devem ser desenvolvidos para direcionar a escolha das operações. No caso de MMD_T, os autores desenvolveram um critério baseado na minimização da distância *Hamming* entre a função especificada e a função identidade, como mostrado na Equação 13 da seção 4.1.1. Porém, quando duas ou mais operações elementares satisfazem o critério de minimização da distância, um segundo critério deve ser considerado. Para este fim, adicionou-se na abordagem proposta MMD *plus* um segundo critério de seleção baseado no custo mínimo: A partir da segunda iteração, se várias operações elementares satisfazem o critério de minimização por distância *Hamming*, a porta reversível ternária de menor custo deve ser escolhida. Na hipótese de várias portas satisfazerem o critério de distância de *Hamming* e tiverem o mesmo custo, então uma escolha aleatória deverá ser feita.

Utiliza-se novamente a função reversível ternária f_1 para demonstrar que a aplicação do critério adicional de seleção reduz o custo total dos circuitos gerados. Na coluna 1 da Tabela 9, duas possibilidades levariam ao vetor de saída desejado [0 0]:

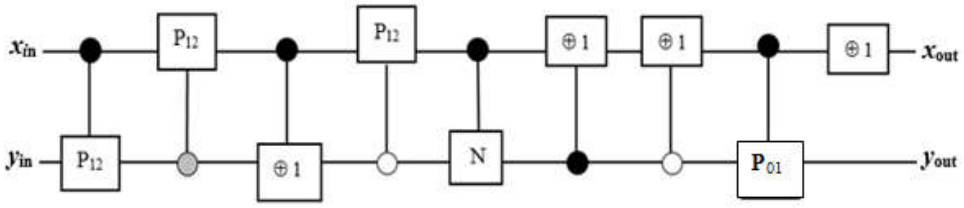
i. $x \oplus 1$.
ii. se $y = 0$, então, $x \oplus 1$

O custo da porta simples ($x \oplus 1$) é igual a 1, e o da porta controlada por 0 é igual a 4, visto que os controles 0 e 1 requerem portas simples auxiliares para realização da operação, e por isso apresenta maior custo. Logo, a porta simples levaria a uma solução menos dispendiosa. Uma vez que ambas as portas reversíveis ternárias satisfizeram o critério de minimização da distância *Hamming,* a escolha foi aleatória. Esta situação também ocorre na coluna da 2, em que três permutações elementares poderiam ser escolhidas:

i. se $x = 1$, então, $P_{12}(y)$;
ii. se $y = 2$, então, $P_{01}(x)$;
iii. se $y = 2$, então, $x \oplus 2$.

As portas controladas por 2 tem o custo igual a 2, porém o algoritmo MMD_T selecionou aleatoriamente a porta controlada por 1 com custo igual a 4. Com a incorporação do segundo critério de seleção ao algoritmo estendido MMD *plus*, seleciona-se a porta de menor custo dentre as portas reversíveis ternárias que produzem a mesma distância *Hamming* entre a função atual e a função identidade. Logo, o custo total do circuito reversível ternário gerado com a incorporação do segundo critério foi reduzido de 28 para 23. Este circuito é apresentado na Figura 8

Figura 8 - Realização da função f_1 com MMD *plus*, modo *backward.*



Fonte: Próprio Autor

Por meio do processo de síntese do algoritmo MMD, exemplificado na Tabela 8, é possível constatar que os vetores de entrada referentes à especificação ternária reversível são ordenados lexicograficamente e, por consequência, os vetores de saída são dispostos de acordo com os de entrada correspondentes. Entretanto, existem m! possibilidades para ordenar os vetores de entrada (m: quantidade de variáveis de entrada/saída), e cada ordenação pode ou não produzir um circuito com complexidade diferente. No domínio reversível, tanto binário quanto ternário, esta situação foi comprovada por Alhagi, Hawash e Perkowski (2012), Hawash e Perkowski (2012) e, recentemente, por Soeken et al. (2016). No trabalho de Hawash e Perkowski (2012) foi demonstrado que a complexidade do circuito pode ser influenciada pela disposição dos vetores de entrada. Os autores ordenaram as entradas de acordo com os conceitos matemáticos do diagrama de Hasse, e mostraram que existem diversas possibilidades para que as entradas sejam dispostas de acordo com o diagrama.

Considerando as caraterísticas apresentadas e as possibilidades de obter circuitos menos complexos, diferentes tipos de ordenamentos para as especificações reversíveis ternárias de entrada são implementadas neste trabalho e, posteriormente, analisadas. As análises de melhoria do circuito são em relação ao custo total e a quantidade de portas reversíveis ternárias dos circuitos sintetizados. Como a quantidade para as possíveis ordenações dos vetores é grande, encontrar a melhor ordenação pode não ser feito de forma eficiente. Em virtude disso, foram escolhidas duas formas de ordenação para os vetores de entrada, que podem levar a circuitos menos dispendiosos. As especificações serão ordenadas por peso e distância *Hamming* entre os vetores de entrada.

4.2.1 Ordenação dos vetores de entrada por distância e peso *Hamming*

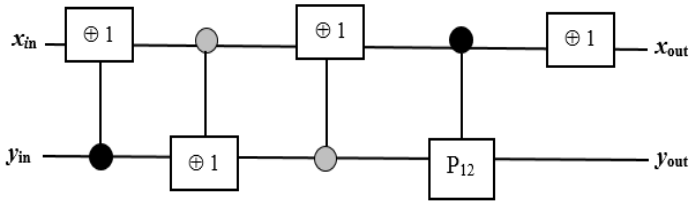
Definição 7: O peso *Hamming d*e um vetor binário $A = \{a_1, \ldots, a_{N-1}\}$ é definido pela quantidade de bits diferentes de zero contidos no vetor, que varia de 0 até *N* (WEI, 1991).

No domínio da síntese reversível binária, sabe-se que ordenar as entradas aumentado o peso *Hamming*, em diversos casos leva a realizações com menor complexidade do que a obtida com o MMD clássico. Uma análise formal foi apresentada por Soeken et al. 2016. Alguns testes realizados preliminarmente no caso ternário, usando um ordenamento análogo ao peso binário *Hamming*, mostraram que realizações com igual ou menor complexidade do que com MMD_T podem ser obtidas.

Utiliza-se novamente da função f_1 para análise. O procedimento de síntese, ordenando os vetores de entrada de acordo com o peso *Hamming*, levou a um circuito reversível ternário

contendo 5 portas e com custo total de 13, ao passo que com o MMD *plus,* cujos os vetores de entrada foram ordenados lexicograficamente, foi gerado um circuito com 9 portas e custo total de 23. O processo de síntese é ilustrado na Tabela 10 e o circuito respectivo na Figura 9. Na primeira coluna da tabela pode-se visualizar como os vetores ternários são ordenados de acordo com peso *Hamming.*

Figura 9 - Realização da função f_1 com MMD *plus*, modo *backward*, em que as entradas são ordenadas de acordo com o peso *Hamming*.



Fonte: Próprio Autor

Tabela 10 - Procedimento de síntese com o MMD *plus* modo *backward*, com as entradas ordenadas de acordo com o peso *Hamming*.

Entrada	Saída	Operações Elementares					
x_{in} y_{in}	x_{out} y_{out}	**1**	**2**	**3**	**4**	**5**	**6**
0 0	2 0	**0 0**	**0 0**	**0 0**	**0 0**	**0 0**	**0 0**
0 1	1 2	2 2	2 1	**0 1**	**0 1**	**0 1**	**0 1**
1 0	0 2	1 2	1 2	1 2	**1 0**	**1 0**	**1 0**
0 2	1 1	2 1	2 2	2 2	2 2	2 2	**0 2**
2 0	1 0	2 0	2 0	2 0	2 0	2 0	**2 0**
1 1	0 0	1 0	1 0	1 0	1 1	1 1	**1 1**
1 2	2 2	0 2	0 2	0 2	0 2	0 2	**1 2**
2 1	0 1	1 1	1 1	2 1	2 1	2 1	**2 1**
2 2	2 1	0 1	0 1	1 1	1 2	1 2	**2 2**

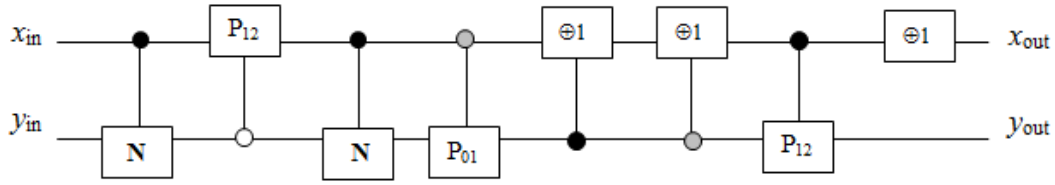
1: $x \oplus 1$; **4**: if $x = 1$, então, $y \oplus 1$;
2: if $x = 2$, então, $P_{12}(y)$; **5**: if $y = 2$, então, $x \oplus 1$.
3: if $y = 1$, então, $x \oplus 1$;

Fonte: Próprio Autor

Para mais uma possível ordenação dos vetores de entrada o comportamento do algoritmo é avaliado, utilizando-se do conceito de ordenação por distância *Hamming*.

Aplicando para o caso ternário, os vetores de entrada se encontram na seguinte disposição [0 0], [0 1], [0 2], [1 2], [1 1], [1 0], [2 0], [2 1] e [2 2]. Esta disposição empregada na função teste f_1 levou a um circuito reversível ternário de 8 portas e com um custo total de 21. O circuito é ilustrado na Figura 10.

Figura 10 - Realização da função f_1 com MMD *plus*, modo *backward*, em que as entradas são ordenadas de acordo com a distância *Hamming*.



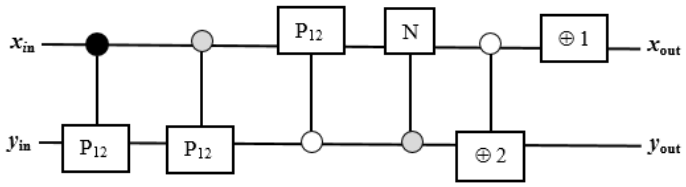
Fonte: Próprio Autor

Portanto, a ordenação dos vetores de entrada na tabela é um fator importante durante o processo de síntese, uma vez que as atribuições de saída estão vinculadas as atribuições de entrada. Além disso, podem levar a uma realização diferente do circuito em relação ao número de portas e custo total.

4.2.2 Orientação do processo de busca em modos *forward, backward, top-down* e *bottom-up*

Em virtude da reversibilidade dos circuitos é igualmente possível trabalhar em modo *backward*, adaptando os vetores de saída aos vetores de entrada, em modo *forward*, adaptando os vetores de entrada aos vetores de saída, e combinando ambos em um modo bidirecional, que é considerado o mais eficiente segundo os autores de MMD_T (MILLER; MASLOV; DUECK, 2006). Os autores demonstraram que os modos *backward*, *forward* e bidirecional geram circuitos com complexidades diferentes. Isto se deve ao fato de que o algoritmo é ganancioso e, em cada iteração, realiza a busca visando um ótimo local. A sequência de ótimos locais não é, necessariamente, a mesma nos diferentes modos. Esta situação pôde ser constatada aplicando a abordagem MMD *plus* no modo *forward* para a função f_1, em que foi obtido um circuito reversível ternário de menor custo e com menor número de portas, quando comparado à realização do algoritmo no modo *backward.* A realização *backward top-down* para a função f_1 encontra-se ilustrada na Figura 8 da seção 4.2 e a *forward top-down* na Figura 11.

Figura 11 - Realização da função reversível ternária f_1 com MMD *plus* modo *forward.*



Fonte: Próprio Autor

Mostra-se que um efeito similar ocorre quando o algoritmo MMD *plus* é aplicado com busca orientada a *bottom-up* e *top-down*. Quando o processo de busca pela solução é iniciado a partir do primeiro vetor de saída define-se que é uma busca *top-down*, além de *backward*. Este tipo de busca é original do algoritmo MMD, sendo definida como o modo padrão de execução. Em alternativa, quando inicia a busca pela solução a partir do último vetor de saída define-se como sendo *bottom-up*.

Realiza-se a função f_1, utilizada como teste, com o algoritmo MMD *plus,* em que o processo de busca é orientando a *backward* e *bottom-up.* Na Tabela 11, este processo é detalhado. A operação inicial é aplicada ao último vetor de saída [2 1]. Posteriormente, uma determinada operação é aplicada ao penúltimo vetor de saída, neste caso, referente à porta reversível ternária NOT. Este processo é realizado sucessivamente até que o primeiro vetor de saída seja igual ao primeiro vetor de entrada e, portanto, tem-se uma função identidade. O circuito resultante é mostrado na Figura 12.

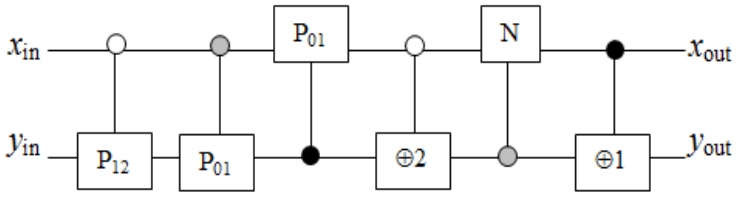
Tabela 11 - Procedimento de síntese com o MMD *plus* modo *backward* e *bottom-up*.

Entrada	Saída	Operações Elementares					
x_{in} y_{in}	x_{out} y_{out}	**1**	**2**	**3**	**4**	**5**	**6**
0 0	2 0	2 1	0 1	0 0	0 0	0 0	**0 0**
0 1	1 2	1 2	1 2	1 2	0 2	0 2	**0 1**
0 2	0 2	0 2	0 2	0 1	0 1	0 1	**0 2**
1 0	1 1	1 1	1 1	1 1	1 1	**1 0**	**1 0**
1 1	1 0	1 0	1 0	1 0	1 0	**1 1**	**1 1**
1 2	0 0	0 0	0 0	0 2	**1 2**	**1 2**	**1 2**
2 0	2 2	2 0	**2 0**	**2 0**	**2 0**	**2 0**	**2 0**
2 1	0 1	0 1	**2 1**	**2 1**	**2 1**	**2 1**	**2 1**
2 2	2 1	**2 2**	**2 2**	**2 2**	**2 2**	**2 2**	**2 2**

1: if $x = 2$, então, $y \oplus 1$; **4**: if $y = 2$, então, $P_{01}(x)$;
2: if $y = 1$, então, $N(x)$; **5**: if $x = 1$, então, $P_{01}(y)$;
3: if $x = 0$, então, $y \oplus 2$; **6**: if $x = 0$, então, $P_{12}(y)$;

Fonte: Próprio Autor

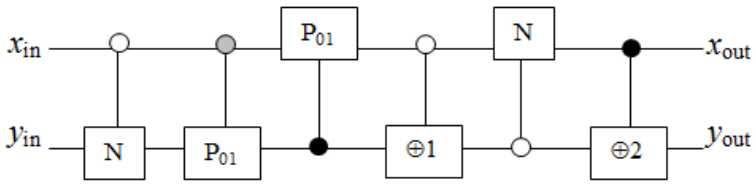
Figura 12 – Realização da função reversível ternária f_1 com MMD *plus* modo *backward* e *bottom-up*.



Fonte: Próprio Autor

Analogamente, a busca orientada a *bottom-up* também é empregado no MMD *plus* modo *forward*. Como mencionado, no modo *forward* os vetores de entradas devem ser adaptados aos vetores de saída. A função f_1 realizada com processo de síntese do MMD *plus* no modo *forward e bottom-up*, resulta em um circuito reversível ternário com 6 portas e com custo total de 20. O circuito é ilustrado na Figura 13.

Figura 13 – Realização da função reversível ternária f_1 com MMD *plus* modo *forward* e *bottom-up*.



Fonte: Próprio Autor

Os resultados obtidos com a abordagem proposta são sumarizados na Tabela 12. É possível constatar que para a função f_1 empregada como teste, o melhor circuito reversível ternário decorreu do processo de busca dos modos *forward* e *top-down.* Contudo, não se pode concluir que para outras especificações reversíveis ternárias, os modos *forward* e *top-down* sempre produzirá circuitos de menor custo e com menor quantidade de portas. Assim sendo, as metodologias propostas para as diferentes ordenações dos vetores de entrada e orientações do processo de busca são testadas exaustivamente para todas 362880 especificações reversíveis ternárias com duas variáveis de entrada/saída. Os resultados dos testes são apresentados no Capítulo 6.

Tabela 12 – Sumarização dos resultados da aplicação dos modos *backward, forward, top-down* e *bottom-up* à função f_1.

Função Reversível Ternária	**Orientação da busca**		**Nº de portas**	**Custo Total**
f_1 = 20,12,11,02, 00,22,10,01,21	*Backward*	*Top-down*	9	23
		Bottom-up	6	20
	Forward	*Top-down*	6	19
		Bottom-up	6	20

Fonte: Próprio Autor

Como mostrado, é igualmente possível trabalhar em modo *backward*, adaptando as saídas às entradas e, em um modo *forward*, adaptando as entradas às saídas. Isso se deve ao fato que todas as operações dos circuitos são logicamente reversíveis. Nos trabalhos de Miller, Maslov e Dueck (2006), os autores demonstraram que o modo bidirecional gerou circuitos de menor complexidade, isto é, com menor custo e menor quantidade de portas reversíveis ternárias nas realizações. Analogamente, realiza-se uma versão bidirecional do MMD *plus,* em que as melhores realizações entre os modos *forward* e *backward* são combinadas.

4.3 MMD *plus* APLICADO A DECOMPOSIÇÃO DE PERMUTAÇÕES EM CICLOS

As propriedades da álgebra abstrata, especificamente a teoria de grupo de permutação, podem ser aplicadas diretamente a síntese de circuitos reversíveis. Isto se deve ao fato de que especificações reversíveis são funções bijetivas, em que os vetores de saída são permutações dos vetores de entrada correspondentes. Empregando-se destes conceitos, qualquer especificação reversível pode ser escrita como uma cascata de ciclos de permutação. De particular interesse tem-se o produto de ciclos de permutação disjuntos, uma vez que estes podem ser reordenados livremente e quando aplicados no processo de síntese permitem a combinação de portas reversíveis ternárias entre ciclos vizinhos, reduzindo, assim, o custo de implementação.

Como mencionado anteriormente, esta abordagem cíclica para a síntese de circuitos reversíveis no domínio binário foi formalizada por Shende et al. (2003), em que especificações reversíveis foram decompostas em ciclos de permutação de tamanho 2 (ciclos com dois elementos) que são denominados de transposições. Posteriormente, Sasanian et al. (2009) provaram que a decomposição de permutações em transposições levou a redundâncias, e, então, introduziram decomposições em ciclos de tamanho 3 e, quando necessário, um ciclo adicional de tamanho 2. Na síntese de circuitos reversíveis ternários uma abordagem semelhante não é conhecida na literatura. Ressalta-se que Yang et al. (2005) comprovaram teoricamente que as propriedades referentes a grupos de permutação podem ser aplicadas diretamente à especificações ternárias, e Li et al. (2013) apresentaram para dois exemplos, de um meio somador e de um somador completo, uma abordagem de síntese construtiva atribuída a ciclos de permutação de tamanho 3. Entretanto, as especificações foram sintetizadas com a adição de entradas constantes e saídas *garbages,* resultando em linhas extras para os circuitos gerados e, por conseguinte, o aumento do custo de implementação.

A presente metodologia foi selecionada para comparar as realizações de todas as 362880 possíveis especificações reversíveis ternárias de duas variáveis com base na aplicação do algoritmo MMD *plus* modo *backward* em ciclos de permutação de ordem natural, em ciclos de permutação decompostos em transposições, transposições compostas em ciclos de tamanho 3.

Nas seções seguintes são apresentadas propriedades importantes para o emprego desta metodologia.

4.3.1 Permutação

Define-se que uma permutação α de um conjunto finito *A* é uma bijeção que leva *A* em *A*. O grupo simétrico do conjunto *A*, denotado por S_A, é o conjunto de todas as possíveis permutações de *A*. Seja $\alpha \in S_n$, então, α é uma permutação em $A = \{1, 2, \ldots, a_{n-1}, a_n\}$, em que *n* representa a quantidade de elementos.

Considerando o conjunto *A* pode-se escrever a permutação correspondente de duas formas. Na primeira forma, um mapeamento $\rho: A \rightarrow A$ é escrito como uma matriz, em que a linha superior contém os elementos do conjunto *A* e a linha inferior representa a imagem de cada elemento sob α, como apresenta-se na Equação 14.

$$\alpha = \begin{pmatrix} 1 & 2 & \cdots & n-1 & n \\ \alpha(1) & \alpha(2) & \cdots & \alpha(n-1) & \alpha(n) \end{pmatrix} \tag{14}$$

Como exemplo, considere a permutação descrita na Equação 15, em que S consiste de 8 elementos:

$$\alpha = \begin{pmatrix} 0\ 1\ 2\ 3\ 4\ 5\ 6\ 7 \\ 1\ 3\ 0\ 7\ 5\ 4\ 2\ 6 \end{pmatrix} = (1,3,7,6,2,0)(4,5) \quad (15)$$

Na segunda forma, uma notação mais compacta é usada para escrever os elementos do conjunto *A*. Esta notação é denominada de cíclica.

4.3.2 Decomposição de permutação em cascata de ciclos disjuntos

Seja α uma permutação de S_n. Associado a α existem os pontos que são fixados e os pontos que são movidos por α.

Definição 8. Os pontos fixos de α são os elementos do conjunto $\{1, 2, \ldots, a_{n-1}, a_n\}$ no qual $\alpha\ (i) = i$.

Definição 9. Os pontos móveis de α são os elementos do conjunto $\{1, 2, \ldots, a_{n-1}, a_n\}$ no qual $\alpha\ (i) \neq i$.

Como exemplo, considera-se a permutação α descrita na Equação 14. Para obter o ciclo de permutação correspondente deve-se escolher um dos elementos para representar o início do ciclo. Neste caso, opta-se pelo elemento 0, gerando o ciclo $c_1 = 0 \rightarrow 1 \rightarrow 3 \rightarrow 7 \rightarrow 6 \rightarrow 2 \rightarrow 0$, em que, atribui-se a 0 ao elemento 1, a 1 o elemento 3, e assim sucessivamente até que o último elemento seja atribuído ao primeiro elemento, encerrando o ciclo. Destaca-se que os elementos 4 e 5 não estão contidos no ciclo c_1, entretanto, estes fazem parte do grupo de permutação e devem ser representados. Neste caso, outro ciclo deve-se formado: $c_2 = 4 \rightarrow 5 \rightarrow 4$.

Como notação alternativa, o ciclo σ_1 de tamanho 6 e o σ_2 de tamanho 2 podem ser escritos como $c_1 = (1,3,7,6,2,0)$ e $c_2 = (4,5)$. Logo, o produto dos ciclos $c_1\ e\ c_2$ representa a permutação α, como mostra-se na Equação 16. Sabe-se, então, que toda permutação pode ser decomposta em uma cascata de ciclos de permutação.

$$c = (1,3,7,6,2,0)(4,5) \quad (16)$$

Dois ciclos de permutação são considerados disjuntos se não existirem elementos comuns entre eles. Como por exemplo, os ciclos de permutação apresentados na Equação 16 são disjuntos.

4.3.3 Produto de transposições

Dada uma permutação e a respectiva notação cíclica, pode-se escrever cada ciclo de permutação como um produto de transposições.

Definição 10: Um ciclo de tamanho 2 (2-ciclo) é considerado uma transposição.

Ademais, qualquer ciclo em S_n com n>1 pode ser escrito como o produto de transposições. Existem diversas formas de decompor um ciclo de permutação. O número de transposições nestas decomposições pode variar. Um ciclo de permutação de tamanho *n* contém no mínimo *n* -1 transposições. Em virtude dessas características, as transposições de um determinado ciclo de permutação não são únicas (BAUMSLAG; CHANDLER, 1968).

Como exemplo, apresentam-se na Equação 17, quatro formas de produto de transposições para o ciclo $(1, 2, \dots, a_{n-1}, a_n)$:

$$
\begin{aligned}
&(1, 2, \dots, a_{n-1}, a_n) = (1,\ a_n)\ (1, a_{n-1}),\quad , (1, 2)\\
&(1, 2, \dots, a_{n-1}, a_n) = (1,2)\ (a_n, a_{n-1}),\quad , (1, a_n)\\
&(1, 2, \dots, a_{n-1}, a_n) = (1, 2)\ (2\ a_{n-1}),\quad , (a_{n-1}, a_n)\\
&(1, 2, \dots, a_{n-1}, a_n) = (2, a_{n-1})\ (2, a_n),\quad , (2, 1)
\end{aligned}
\tag{17}
$$

Considerando o produto de ciclos disjuntos $(0, 1, 3, 7, 6, 2)$ (4,5), obtidos na Equação 15, o primeiro ciclo $(0, 1, 3, 7, 6, 2)$ pode ser decomposto resultando no seguinte produto de transposições (0, 1) (0, 3) (0, 7) (0, 6) (0, 2). Cabe ressaltar que, o ciclo (4, 5) encontra-se na forma de transposição, pois possui dois elementos e, portanto, não precisa ser decomposto.

4.3.4 Composição de Transposições em Ciclos com 3 elementos (ciclos – 3)

Deve-se ressaltar que existem duas propriedades relativas às permutações em cascata que são importantes no contexto abordado. Permutações consideradas como elementos de um

grupo cuja operação é um produto, levando a aplicar as permutações na ordem em que se apresentam, da esquerda para a direita. Por outro lado, as permutações são funções e sua cascata pode ser entendida como composição. Se α_1 e α_2 são permutações em um conjunto S_n e $i \in S_n$, então $\alpha 1 \alpha 2(i) = (\alpha_1 \circ \alpha_2)(i) = \alpha_1(\alpha_2(i))$, ou seja, as permutações em cascata serão aplicadas da direita para a esquerda . No que se segue, a alternativa de composição será usada com alternativa de síntese, como mostrado na Equação 18. Dada uma determinada permutação, realiza-se a decomposição em produto de transposições. Em seguida, as transposições são reduzidas a ciclos de tamanho 3, por meio de composições. Se a quantidade de transposições decompostas for ímpar, por consequência restará uma única transposição. A transposição remanescente deve ser incluída com os ciclos de 3 elementos que representam a permutação dada.

$$(A,B,C,D,E) = (A,E)(A,D)(A,C)(A,B) = (A,D,E)(A,B,C) \quad (18)$$

4.3.5 MMD *plus* aplicado a ciclos de permutação de ordem natural

A princípio, alguns aspectos importantes para a estrutura específica dos circuitos reversíveis ternários realizados a partir de ciclos de permutação devem ser considerados:

I. Uma cascata de ciclos disjuntos pode ser reordenada. Se os ciclos não forem disjuntos, a ordem da especificação deve ser preservada;

II. A representação completa de um ciclo de permutação é obtida aplicando o ciclo aos vetores de entradas. Os vetores de entrada que não são afetados por um ciclo mantêm seus valores;

III. A cascata é processada como composição de funções: da direita para a esquerda. Do ponto de vista do circuito, no entanto, isso significa que um primeiro subcircuito deve calcular o efeito do ciclo mais à direita. Isto pode ser estendido de forma recursiva para qualquer número de ciclos em uma cascata;

IV. O circuito final compreenderá uma sequência de subcircuitos ordenados de acordo com a ordem inversa dos ciclos.

Para análise das alternativas de síntese designa-se uma determinada função reversível ternária f_2 de duas variáveis na forma de tabela de valores, especificada na
Tabela 13.

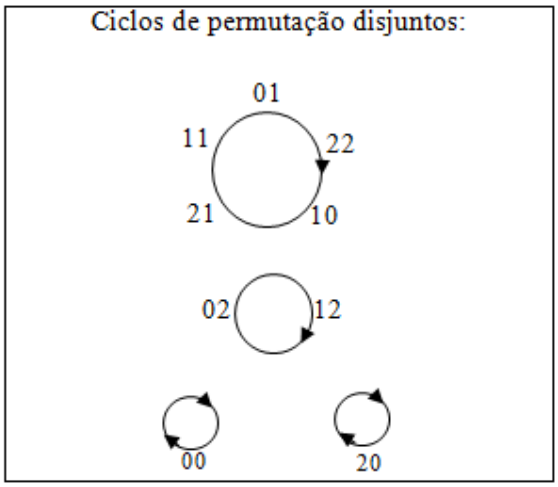
Tabela 13 – Especificação da função reversível ternária f_2: (00, 22, 12, 21, 01, 02, 20, 11, 10)

Entrada		Saída	
x_{in}	y_{in}	x_{out}	y_{out}
0	0	0	0
0	1	2	2
0	2	1	2
1	0	2	1
1	1	0	1
1	2	0	2
2	0	2	0
2	1	1	1
2	2	1	0

Fonte: Próprio Autor

A função f_2 representada na forma de ciclo de permutação resume-se em um produto de dois ciclos disjuntos (01, 22, 10, 21, 11) (02, 12). Os ciclos com apenas um elemento não requerem representação e por isso são omitidos. A decomposição é ilustrada na Figura 14

Figura 14 - Ilustração da decomposição da função reversível ternária f_2 em produto de ciclos de permutação disjuntos.

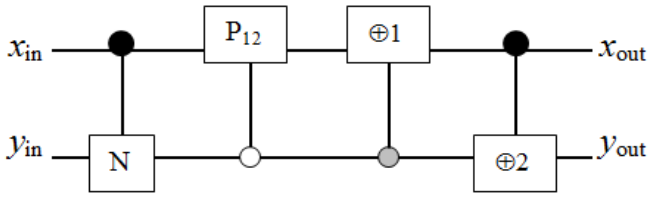


Fonte: Próprio Autor

O processo de síntese deve ser aplicado individualmente a cada ciclo de permutação. As operações de síntese com base no MMD *plus* empregadas ao primeiro ciclo (01, 22, 10, 21, 11) são mostradas na Tabela 14. Os vetores de entrada e de saída que estão sombreados correspondem aos elementos do ciclo a ser sintetizado. Os demais permanecem fixos, pois não

pertencem a tal ciclo. Obteve-se um circuito reversível ternário com 4 portas reversíveis ternárias e com um custo total de 12, como pode ser visto na Figura 15.

Figura 15 – Circuito gerado com a aplicação de MMD *plus* ao ciclo de permutação (01, 22, 10, 21, 11).



Fonte: Próprio Autor

Tabela 14 – Algoritmo MMD *plus* backward top-down aplicado ao ciclo de permutação (01, 22, 10, 21, 11).

Entrada	Saída	Operações Elementares			
x_{in} y_{in}	x_{out} y_{out}	**1**	**2**	**3**	**4**
0 0	0 0	**0 0**	**0 0**	**0 0**	**0 0**
0 1	2 2	2 1	**0 1**	**0 1**	**0 1**
0 2	0 2	0 2	**0 2**	**0 2**	**0 2**
1 0	2 1	2 0	2 0	**1 0**	**1 0**
1 1	0 1	0 1	1 1	**1 1**	**1 1**
1 2	1 2	1 2	1 2	**1 2**	**1 2**
2 0	2 0	2 2	2 2	2 2	**2 0**
2 1	1 1	1 1	2 1	2 1	**2 1**
2 2	1 0	1 0	1 0	2 0	**2 2**

1: if $x = 2$, então, $x \oplus 2$; **3**: if $y = 0$ então $P_{12}(x)$;

2: if $y = 1$, então, $x \oplus 1$; **4**: if $x = 2$, então, $N(y)$;

Fonte: Próprio Autor

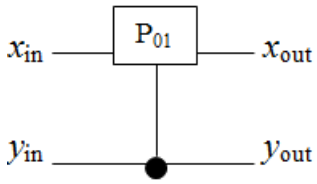
Igualmente, aplica-se o algoritmo do MMD *plus* ao segundo ciclo de permutação (02, 12). As operações de síntese são mostradas na Tabela 15. Obteve-se um circuito reversível ternário com 1 porta reversível ternária e com um custo igual a 2, como pode ser visto na Figura 16.

Tabela 15 - Algoritmo MMD *plus* aplicado ao ciclo de permutação (02, 12).

Entrada	Saída	Operação Elementar
x_{in} y_{in}	x_{out} y_{out}	**1**
0 0	0 0	0 0
0 1	0 1	0 1
0 2	1 2	0 2
1 0	1 0	1 0
1 1	1 1	1 1
1 2	0 2	1 2
2 0	2 0	2 0
2 1	2 1	2 1
2 2	2 2	2 2
1: Se $y = 2$,então, $P_{01}(x)$;		

Fonte: Próprio Autor

Figura 16 - Circuito gerado com a aplicação de MMD *plus* ao ciclo de permutação (02, 12).

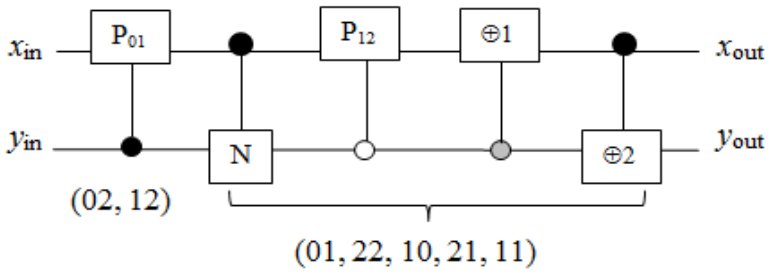


Fonte: Próprio Autor

Portanto, para cada ciclo de permutação, decomposto a partir da função f_2, obteve-se uma representação completa na forma de tabela de valores, em que o algoritmo de síntese MMD *plus* foi empregado. A sequência de subcircuitos originados por meio da síntese sobre cada tabela compreende o circuito final. Como os ciclos do exemplo são disjuntos, uma ordenação da cascata de subcircuitos pode ser escolhida para favorecer a possível mesclagem de portas vizinhas entre dois ciclos. Duas portas que contém o mesmo valor de controle e são dispostas paralelamente são consideradas vizinhas, e quando mescladas resultam em uma única porta, minimizando o custo total do circuito. Para o circuito resultante da síntese empregado aos ciclos de permutação esta situação não ocorre. Contudo, o circuito possui 5 portas reversíveis ternárias com custo total de 14, enquanto que para a mesma função f_2 sintetizada diretamente com MMD *plus backward top-down*, isto é, sem a decomposição em ciclos, o circuito contém 8 portas e

um custo total igual a 21. Portanto, para este exemplo a decomposição em ciclos foi vantajosa gerando um circuito menos dispendioso, mesmo sem a mesclagem entre portas após a síntese.

Figura 17 – Circuito reversível ternário correspondente aos ciclos de permutação (01, 22, 10, 21, 11) (02, 12).



Fonte: Próprio Autor

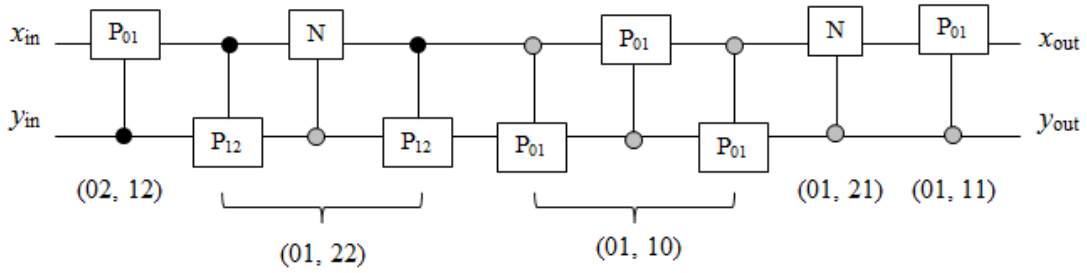
4.3.6 MMD *plus* aplicado a transposições

De forma análoga a metodologia descrita no Exemplo 1, da seção 4.3.4, realiza-se a síntese sobre cada ciclo de permutação com 2 elementos, que são denominados de transposições. As transposições são obtidas a partir da decomposição dos ciclos de permutação. Como mencionado, existem diferentes formas de decomposição (BAUMSLAG; CHANDLER, 1968). Seguindo a decomposição proposta por Sedgewick (1977), os ciclos (01, 22, 10, 21, 11) (02, 12) obtidos para a função f_2 podem ser decompostos nas transposições descritas na Equação 19.

$$\begin{aligned} f_2 &= (01, 22, 10, 21, 11)\ \mathbf{(02, 12)} \\ f_2 &= (01, 11)\ (01, 21)\ (01, 10)\ (01, 22)\ \mathbf{(02, 12)} \end{aligned} \tag{19}$$

Seguidamente a decomposição, cada transposição é sintetizada com o algoritmo MMD *plus*. Os subcircuitos gerados compreendem o circuito final. Ressalta-se que como as transposições não são ciclos de permutação disjuntos, a sequência de subcircuitos deve ser preservada. Na Figura 18 o circuito reversível ternário que corresponde às transposições é apresentado. Sabe-se, então, que as transposições (01, 21) e (01, 11) têm um custo individual igual a 4, a transposição (01, 10) um custo igual a 12, a transposição (01, 22) um custo igual a 8 e, por fim, a transposição (02, 12) um custo igual a 2, resultando no custo total igual 30.

Figura 18 - Circuito reversível ternário correspondente às transposições (01, 11) (01, 21) (01, 10) (01, 22) (02, 12).



Fonte: Próprio Autor

Se um processamento pós-síntese for considerado, mesclando portas reversíveis vizinhas é possível reduzir o custo total e a quantidade de portas do circuito. As portas NOT e P_{01} que realizam as transposições (01, 21) e (01, 11) podem ser mescladas em uma única porta, como mostra-se na Equação 20. As matrizes referentes às portas NOT e P_{01} são multiplicadas, resultando na matriz da porta $\oplus 2$. Dessa forma, as portas NOT e P_{01} são substituídas pela porta $\oplus 2$, reduzindo a quantidade de portas de 9 para 8 e o custo total do circuito de 30 para 26. Mesmo com a mesclagem o custo total do circuito é maior, se comparado a realização do MMD *plus* aos ciclos de permutação de ordem natural, que possui um custo igual 14.

$$\mathrm{NOT} * \mathrm{P}_{01} = \oplus 2$$

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{20}$$

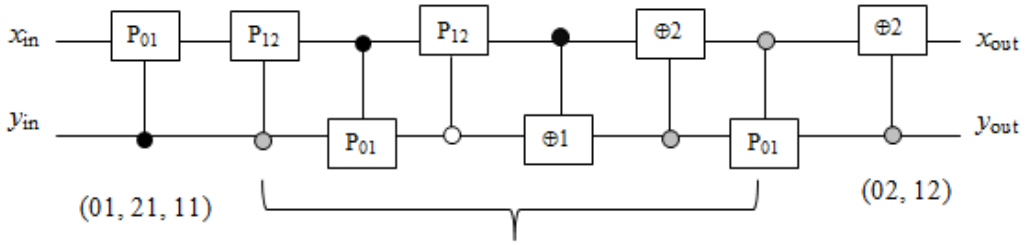
É importante ressaltar que a decomposição de uma permutação em transposições origina um conjunto de ciclos não disjuntos. Portanto, transposições não podem ser reordenadas. Dependerá se as portas vizinhas podem ser mescladas ou não, para possivelmente reduzir o custo de realizações. O exemplo ilustra esta situação mostrando que apenas uma mesclagem foi possível.

4.3.7 MMD *plus* aplicado a ciclos com 3 elementos (ciclos-3)

Como alternativa de síntese, as transposições são reduzidas a ciclos-3 por meio de composições e se necessário uma transposição adicional. Os ciclos obtidos são apresentados na Equação 21, e o circuito reversível ternário a partir da aplicação de MMD *plus* a cada ciclo é mostrado na Figura 19.

$$f_2 = (01, 11)\ (01, 21)\ (01, 10)\ (01, 22)\ \mathbf{(02, 12)}$$
$$f_2 = (01, 21, 11)\ (01, 22, 10)\ \mathbf{(02, 12)} \qquad (21)$$

Figura 19 - Circuito reversível ternário correspondente aos ciclos (01, 21, 11) (01, 22, 10) **(02, 12).**



Fonte: Próprio Autor

Enquanto que o processo de síntese do MMD *plus* aplicado aos ciclos de permutação de ordem natural gerou um circuito reversível ternário com 5 portas e com um custo total de 14, para a realização com as transposições e ciclos-3 foram obtidos circuitos com 8 portas e custo total de 26. Deve ficar claro que o fato de que, para a função f_2 utilizada nestes exemplos, a decomposição em transposições e a composição em ciclos-3 não foram favoráveis, não implica que para outras funções reversíveis ternárias esta situação sempre ocorra. Outro aspecto que deve ser ressaltado é o fato de que somente os subcircuitos obtidos com as transposições permitiram a mesclagem de portas posteriormente a realização da síntese do MMD *plus*. Sem o emprego da abordagem pós-síntese o circuito resultante é composto por 9 portas com um custo total de 30. Portanto, para este exemplo, a metodologia que emprega a síntese a partir de transposições é a de maior custo, e a síntese realizada sobre ciclos de permutação de ordem natural é a de menor custo e quantidade de portas. Os custos e quantidade de portas obtidos em cada metodologia são apresentados na Tabela 16.

Os resultados comparativos sobre as 9! especificações reversíveis ternárias de duas variáveis serão apresentados no capítulo 6.

Tabela 16 - Custos dos circuitos reversíveis ternários obtidos a partir das metodologias I, II e III empregadas à função $f_2 = (00, 22, 12, 21, 01, 02, 20, 11, 10)$.

Metodologias de Síntese	Custos	Nº de portas
MMD *plus* (00, 22, 12, 21, 01, 02, 20, 11, 10)	21	8
MMD *plus* – Ciclos de permutação natural (01, 22, 10, 21, 11) (02, 12)	14	5
MMD *plus* - Transposições (01, 11) (01, 21) (01, 10) (01, 22) (02, 12)	26	8
MMD *plus* – Ciclos-3 (01, 21, 11) (01, 22, 10) (02, 12)	26	8

Fonte: Próprio Autor

5 METODOLOGIA PARA A SÍNTESE DE CIRCUITOS REVERSÍVEIS TERNÁRIOS COM BASE NO ALGORITMO GENÉTICO SIMPLES

Para resolver os problemas associados a uma estrutura não identificada, pesquisadores têm se voltado cada vez mais para os métodos referentes a *Soft Computing*, especialmente os Algoritmos Genéticos (AGs), que são uma das abordagens para a resolução de problemas pertencentes aos Algoritmos Evolutivos (GOLDBERG, 1989).

Os AGs são técnicas de busca fundamentadas nos princípios de seleção e evolução natural do indivíduo (Goldberg, 1989). Uma determinada população inicial de possíveis soluções evolui até convergir para uma solução considerada ideal, por meio da aplicação de operadores genéticos de seleção, cruzamento e mutação. Tais características os tornam viáveis à síntese de circuitos reversíveis ternários utilizando-se das portas reversíveis ternárias Muthukrishnan-Stroud (KHAN et al., 2007; KHAN, 2008). Encontrar uma estrutura ideal é de difícil abordagem, uma vez que a estrutura de uma determinada cascata de portas não é conhecida incialmente e o espaço de busca aumenta exponencialmente de acordo com a quantidade de variáveis de entradas.

A aplicação dos AGs para construir circuitos reversíveis e quânticos já foi investigada (DEBUIK, 2015). Entretanto, obter circuitos reversíveis ternários mínimos continua sendo um desafio. Em particular, isto aplica-se a parâmetros, como por exemplo, o custo total dos circuitos sintetizados, o custo quântico, o número de entradas constantes, dentre outros. O problema é escolher uma estratégia evolutiva que garanta a busca da melhor solução e, simultaneamente, seja processada em um tempo de execução aceitável. Uma estratégia eficiente está diretamente relacionada com a codificação dos indivíduos, a geração da população inicial e as formas efetivas de implementar os principais operadores genéticos (seleção, cruzamento, mutação). A tarefa é ainda mais complicada pelas condições adicionais impostas pelas propriedades de reversibilidade.

Dessa forma, encontrar uma solução adequada ou ótima é um problema exponencialmente grande, não sendo factível usar qualquer método determinístico ou direto. Em virtude disso, utiliza-se do AG que é capaz de pesquisar em um amplo espaço de solução dentro de um período de tempo razoável.

5.1 ALGORITMO GENÉTICO SIMPLES PROPOSTO

Nesta seção é apresentado um processo de síntese a partir de especificações reversíveis ternárias com base no AGS. Propõe-se uma abordagem prática para sintetizar diretamente utilizando o conjunto de portas reversíveis ternárias definidas neste trabalho. Com o emprego do AGS encontra-se uma combinação apropriada de portas que realizem qualquer função reversível ternária de duas variáveis. O principal objetivo do algoritmo é determinar soluções com menos portas reversíveis do que as determinadas pelo MMD *plus,* para as mesmas especificações reversíveis ternárias de entrada.

Diferentes aspectos do algoritmo proposto são apresentados nas seções subsequentes.

5.1.1 Codificação do problema

A solução candidata a resolução de um determinado problema é representada por um indivíduo nos AGs. Encontrar a melhor representação para a solução candidata de acordo com o problema é sempre desejável. Dentre as diversas formas de representar um indivíduo, a representação binária de tamanho fixo, em que um individuo é uma cadeia de bits que assumem valores 0 ou 1, é a mais simples e comumente usada. No AGS proposto utiliza-se uma representação particular para cada indivíduo. O individuo é codificado como um vetor de *strings*, e cada um destes representa uma possível solução para o problema da síntese, isto é, um circuito reversível ternário. Logo, cada elemento do vetor caracteriza uma porta reversível ternária, como pode ser visto na Figura 20.

Figura 20 – Tipo de codificação utilizada para representar cada indivíduo.

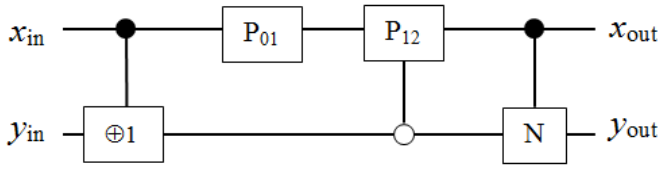
x, C2, X	*x*, F, P01	*y*, C1, P12	*x*, C2, N

Fonte: Próprio Autor

Uma vez que os vetores são indivíduos, cada elemento do vetor é definido como um gene. A primeira variável de cada gene é sempre *x* ou *y*, designando se o controle da porta reversível ternária encontra-se na primeira linha (*x*) ou na segunda (*y*). A segunda variável indica se a porta é controlada por 0, 1, 2 ou se não possui controle. Os controles 0, 1 e 2 são representeados por C0, C1, e C2, respectivamente, e a porta sem controle pelo caractere ‘F’. Por fim, a terceira

variável de cada elemento do vetor designa umas das portas reversíveis ternárias. Os caracteres X, Xt, P01, P12 e N referem-se às portas ⊕1, ⊕2, $\mathbf{P}_{01,}$ $\mathbf{P}_{12,}$ e NOT, respectivamente. Na Figura 21, apresenta-se o circuito que corresponde ao indivíduo exemplificado na Figura 20.

Figura 21 – Circuito reversível ternário equivalente ao indivíduo da Figura 20.



Fonte: Próprio Autor

5.1.1.1 Tamanho dos indivíduos

No AGS proposto, duas abordagens distintas são desenvolvidas para definir o tamanho dos indivíduos. Denominam-se tais abordagens de Menos-Um e Randômica.

Para a avaliação do AGS, o método MMD *plus* será utilizado como parâmetro de comparação. Dessa forma, não é interessante gerar soluções inferiores, consistindo o objetivo principal em reduzir a quantidade de portas dos circuitos. Na abordagem Menos-Um, como o próprio nome sugere, a população inicial é gerada com indivíduos (circuitos) com menos uma porta reversível ternária do que a solução obtida com base no MMD *plus* para a mesma especificação.

Na abordagem Randômica, os tamanhos dos indivíduos são determinados aleatoriamente dentre um tamanho mínimo e máximo predefinidos. Os tamanhos mínimos e máximos são atribuídos as variáveis l_{min} e $l_{máx}$. Define-se que o l_{min} para os indivíduos é de 2, visto que circuitos reversíveis ternários com menos de 2 portas não seriam encontrados para a maior parte das funções. O $l_{máx}$ definido é a quantidade de portas da solução gerada pelo MMD *plus* menos uma porta, pois se busca por uma solução melhor que a encontrada por MMD *plus*. Sendo assim, considerando o intervalo fechado $[l_{min}, l_{máx}]$, um número aleatório $l \in [l_{min}, l_{máx}]$ é selecionado. Logo, o AGS será executado n vezes, sendo $l \leq n \leq b$.

5.1.2 Geração da população inicial

Um conjunto de indivíduos constitui uma população nos AGs. Assim sendo, uma população é um conjunto de possíveis soluções para um determinado problema. Comumente, a população inicial com as possíveis soluções é gerada randomicamente. Existem algumas heurísticas para a inserção de indivíduos previamente conhecidos na população. O importante na geração da população é garantir a diversidade dos indivíduos, ou seja, garantir que os indivíduos estejam distribuídos homogeneamente por todo o espaço de busca. O tamanho da população indica a quantidade de indivíduos. Dessa forma, a diversidade de soluções acresce de acordo com o tamanho da população, bem como o custo computacional também acresce em função da quantidade de soluções para serem avaliadas. Portanto, o tamanho da população influencia diretamente na eficiência dos AGs.

Para resolver o problema da síntese de circuitos reversíveis ternários utilizando-se do AGS, a população inicial foi construída randomicamente, uma vez que o algoritmo MMD *plus* utilizado como parâmetro de comparação dos resultados, gera soluções sem conhecimento prévio.

5.1.3 Função de aptidão

Uma das caraterísticas mais importantes dos AGs é a avaliação de cada indivíduo por meio da função de aptidão. Tal função influencia diretamente na convergência do processo de busca pela solução desejada. Para cada indivíduo avaliado é atribuído um valor de aptidão, representando a qualidade do circuito codificado (indivíduo). No AGS proposto utiliza-se uma forma particular para calcular a função de aptidão, com a finalidade de otimizar os circuitos, reduzindo a quantidade de portas reversíveis ternária dos circuitos em relação aos gerados pelo MMD *plus*. Neste contexto, o componente de aptidão fundamental para a síntese utilizando-se do AGS é a distância *Hamming*.

Para determinar o valor de aptidão, inicialmente especifica-se uma função reversível ternária empregada como entrada do AGS. O AGS percorre todo o espaço de busca com a finalidade de encontrar um circuito que corresponde exatamente à especificação. Esta condição é verificada por uma tautologia da função de especificação de entrada e da função de solução, que consiste no circuito encontrado. No contexto de um circuito reversível com realizações quânticas, a verificação pode ser feita por meio da função de transferência, comparando matrizes unitárias, como mostrado na seção 3.3.1. No caso de funções de permutação, como as

funções reversíveis ternárias, pode-se efetuar a verificação por meio da Distância *Hamming*, comparando as tabelas de valores da função de especificação e de solução. O cálculo que define a Distância *Hamming* entre duas permutações/especificações foi descrito na Equação 13 da seção 4.1. Isto posto, considera-se que o valor de Distância *Hamming* ideal é zero, indicando que não existem diferenças entre a função especificada como entrada e a função de solução e, portanto, o circuito gerado pelo AGS corresponde exatamente a função especificada.

5.1.4 Seleção

Os AGs são fundamentados no princípio de seleção natural e, por consequência, devem ser capazes de identificar os indivíduos mais aptos, para que sobrevivam no processo de evolução da população, e os mais frágeis, para que sejam excluídos do processo (DEB, 2001). A melhoria da população é consequência da repetida seleção dos indivíduos mais aptos, que tem maior chance de produzir bons descendentes (FONSECA, 1995).

Existem diversos métodos utilizados para a seleção dos indivíduos que possuem melhor função de aptidão, como por exemplo, os métodos por torneio, roleta, amostragem determinísticos, dentre outros. No AGS proposto utilizam-se dois métodos de seleção: Elitismo e Proximidade de Inteiros.

5.1.4.1 Elitismo

No processo de seleção por elite, a estratégia consiste em selecionar os *n* melhores indivíduos da população inicial e preservá-los para a próxima geração, garantindo também que o melhor indivíduo seja mantido durante o processo evolutivo. Na Figura 22, ilustra-se uma população inicial com 7 indivíduos. Para cada indivíduo, a função de aptidão é calculada seguindo a metodologia descrita na seção 5.3. Sabe-se que a função de aptidão ideal é igual a zero. Dessa forma quanto menor o valor da função de aptidão do indivíduo, mais apto estará para ser um descendente e permanecer na próxima geração.

Figura 22 – Representação de uma possível população inicial.

População Inicial	*x*, C2, X	*y*, C1, P01	*y*, C0, P12	*x*, F, N	3
	y, F, X	*x*, C2, N	*y*, C0, Xt	*x*, C2, P12	8
	y, C0, X	*y*, C2, X	*x*, C1, N	*y*, C0, N	5
	x, C1, X	*y*, C0, P01	*y*, C2, P12	*x*, F, Xt	10
	x, F, X	*y*, C1, N	*x*, C2, Xt	*x*, F, P01	14
	y, C1, X	*x*, C2, P01	*y*, C1, N	*x*, C0, X	9
	x, F, X	*x*, C2, N	*y*, C1, P12	*x*, C2, P12	7

Fonte: Próprio Autor

De acordo com o valor da função de aptidão de cada indivíduo, a população é reordenada decrescentemente, como mostra-se na Figura 23. Logo, o ultimo indivíduo é o mais apto da população atual e é então preservado para que seja mantido na próxima geração.

Figura 23 – Tipo de reordenamento aplicado a populações iniciais.

População Inicial Reordenada	*x*, F, X	*y*, C1, N	*x*, C2, Xt	*x*, F, P01	14
	x, C1, X	*y*, C0, P01	*y*, C2, P12	*x*, F, Xt	10
	y, C1, X	*x*, C2, P01	*y*, C1, N	*x*, C0, X	9
	y, F, X	*x*, C2, N	*y*, C0, Xt	*x*, C2, P12	8
	x, F, X	*x*, C2, N	*y*, C1, P12	*x*, C2, P12	7
	y, C0, X	*y*, C2, X	*x*, C1, N	*y*, C0, N	5
	x, C2, X	*y*, C1, P01	*y*, C0, P12	*x*, F, N	3

Fonte: Próprio Autor

Com exceção do indivíduo mais apto que foi preservado inicialmente, os $n/2$ melhores indivíduos são copiados e duplicados para a próxima geração. Esta situação é ilustrada na Figura 24.

Figura 24 – População descendente.

População Descendente	*x*, C2, X	*y*, C1, P01	*y*, C0, P12	*x*, F, N	3
	y, F, X	*x*, C2, N	*y*, C0, Xt	*x*, C2, P12	8
	x, F, X	*x*, C2, N	*y*, C1, P12	*x*, C2, P12	7
	y, C0, X	*y*, C2, X	*x*, C1, N	*y*, C0, N	5
	y, F, X	*x*, C2, N	*y*, C0, Xt	*x*, C2, P12	8
	x, F, X	*x*, C2, N	*y*, C1, P12	*x*, C2, P12	7
	y, C0, X	*y*, C2, X	*x*, C1, N	*y*, C0, N	5

Fonte: Próprio Autor

Portanto, a população descendente contém os indivíduos considerados mais aptos para a solução do problema e, por isso, sobreviveram no processo.

5.1.4.2 Proximidade de Inteiros

Além do emprego do elitismo para a seleção dos indivíduos que possuem melhor função de aptidão, desenvolveu-se um novo tipo de seleção para avaliar possibilidades de melhorias na convergência do algoritmo, que consiste na busca por soluções consideradas ideais. Denomina-se este novo tipo de seleção de Proximidade de Inteiros.

Para exemplificar como é realizada a seleção por meio da Proximidade de Inteiros, utiliza-se a mesma população inicial exemplificada na Figura 22. Para cada indivíduo a função de aptidão é calculada seguindo a metodologia descrita na seção 5.1.3. Sabe-se que a função de aptidão ideal é igual a zero. Dessa forma quanto menor o valor da função de aptidão do indivíduo, mais apto estará para ser um descendente e permanecer na próxima geração.

Na Tabela 17, a primeira coluna refere-se aos indivíduos da população inicial gerada randomicamente, e a segunda a função de aptidão de cada indivíduo. Na terceira coluna é calculada a proximidade entre a solução atual e a solução desejável. Sabe-se que a maior distância *Hamming* entre duas soluções é igual a 18, assim o valor da função de aptidão é subtraído de 18, obtendo o valor de proximidade. Portanto, para o primeiro indivíduo o valor de proximidade é 15. Quanto maior for o valor de proximidade entre a solução atual e a desejada, maior será a chance de sobrevivência do indivíduo. Realiza-se a média ponderada dos

valores de proximidade, que é igual a 9,42 para este exemplo. Os valores de proximidade de todos os indivíduos são divididos pela média ponderada, gerando a quarta coluna designada de Proximidade de Inteiros. Para o primeiro indivíduo tem-se que $\frac{15}{9,42} = 1,66$. O número inteiro mais próximo deste valor corresponde à quantidade de vezes em que o indivíduo estará contido na população descendente.

Tabela 17- Seleção dos indivíduos por meio da Proximidade de Inteiros.

INDIVÍDUOS				**F. A**	**Prox.**	**Prox. por inteiros**	**Nº de indivíduos**
x, C2, X	*y*, C1, P01	*y*, C0, P12	*x*, F, N	3	15	1,66	2
y, F, X	*x*, C2, N	*y*, C0, Xt	*x*, C2, P12	8	10	1,11	1
y, C0, X	*y*, C2, X	*x*, C1, N	*y*, C0, N	5	13	1,44	1
x, C1, X	*y*, C0, P01	*y*, C2, P12	*x*, F, Xt	10	8	0,88	1
x, F, X	*y*, C1, N	*x*, C2, Xt	*x*, F, P01	14	4	0,44	0
y, C1, X	*x*, C2, P01	*y*, C1, N	*x*, C0, X	9	9	1	1
x, F, X	*x*, C2, N	*y*, C1, P12	*x*, C2, P12	7	11	1,22	1

Fonte: Próprio Autor.

5.1.5 Operadores genéticos

O princípio básico dos operadores genéticos é transformar a população por meio de sucessivas gerações, estendendo a busca até que seja encontrado um resultado satisfatório. Os operadores genéticos são necessários para que a população se diversifique e mantenha características de adaptação adquiridas pelas gerações anteriores. Os operados genéticos utilizados são os de recombinação e mutação.

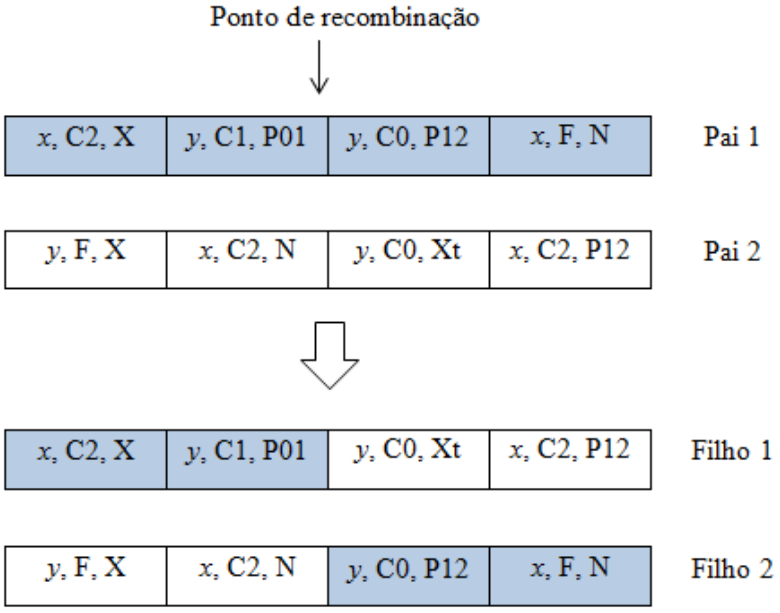
5.1.5.1 Recombinação

O processo de recombinação é realizado posteriormente ao de seleção. É também conhecido na literatura especializada como cruzamento. Neste processo é realizada a troca de material genético entre dois indivíduos denominados pais, criando novas soluções, ou seja, os indivíduos filhos. Espera-se que se bons genes dos pais forem combinados, os indivíduos filhos provavelmente terão melhoras na função de aptidão, criando uma melhor descendência. Ainda assim, o efeito do cruzamento pode ser prejudicial ou benéfico. Portanto, para preservar alguns

dos indivíduos pais considerados bons, nem todos são usados no processo e a recombinação é aplicada com uma determinada probabilidade a cada par de indivíduos selecionados. Esta probabilidade é denominada de taxa de recombinação, variando em média entre 50% e 100%. Se a recombinação não for realizada, os indivíduos filhos serão iguais aos pais, e algumas soluções são preservadas.

Existem diversos tipos de recombinação. O tipo utilizado no algoritmo proposto é a recombinação de um único ponto (*single point crossing over*), em que se escolhe aleatoriamente um ponto/gene do indivíduo e a troca de material genético é realizada a partir do ponto sorteado em diante. Na Figura 25, mostra-se um exemplo de dois indivíduos pais (azul e branco) recombinados por um único ponto para gerar dois indivíduos filhos.

Figura 25 – Processo de recombinação entre dois indivíduos país.



Fonte: Próprio Autor

5.1.5.2 Mutação

A mutação é um processo que possibilita a inserção de novas características genéticas em indivíduos durante o processo de evolução, possibilitando maior diversidade populacional. É efetuada alterando, eventualmente, aspectos da solução atual. A taxa de mutação indica a probabilidade em que haverá mutação nos indivíduos ao longo da evolução. O *crossover* visa recombinar partes significativas dos indivíduos pais para criar uma melhor descendência. A

mutação, por outro lado, altera uma única *string* localmente do filho para criar um filho superior.

5.1.6 Critério de parada

Posteriormente a aplicação dos operadores genéticos de recombinação e mutação, realiza-se a avalição dos critérios de parada do processo de busca pela solução. No AGS proposto foram utilizados dois critérios de parada. Quando o valor ideal da função de aptidão for encontrado e/ou quando uma quantidade máxima de gerações estabelecida é atingida. Os passos que descrevem a busca usando o AGS são chamados de gerações.

5.1.7 Descrição do algoritmo genético simples para a síntese de circuitos reversíveis ternários

As etapas do AGS modificado para a síntese de circuitos reversíveis ternários consistem em:

1. Uma população aleatória de indivíduos é determinada. Cada indivíduo é um circuito reversível ternário. O tamanho da população varia entre 50 e 200 indivíduos. O tamanho do indivíduo é definido de acordo com a abordagem utilizada, MENOS-UM ou Randômica;

2. Cada indivíduo é avaliado por meio da função de aptidão descrita na 5.1.3;

3. Os critérios de parada são analisados. Se forem satisfeitos, o processo de busca pela solução é interrompido. Senão a próxima etapa deve ser executada;

4. Seleção dos melhores indivíduos para gerar a população descendente. As regras de seleção de indivíduos podem acelerar a convergência da pesquisa. No entanto, se o operador de seleção for muito ganancioso, selecionando apenas os melhores indivíduos, o algoritmo pode ficar estagnado em um mínimo local. Utilizam-se os métodos de seleção por Elite, descrito na seção 5.1.4.1, e de seleção por Proximidade de Inteiros, definido na seção 5.1.4.2;

5. Para a população descendente, que contém os indivíduos filhos mais apta à solução, são aplicados os operadores genéticos de Recombinação e Mutação. A taxa de recombinação

T_R utilizada é de 50%. Na seção 5.2.1.1, descreve-se com detalhes como a recombinação é aplicada.

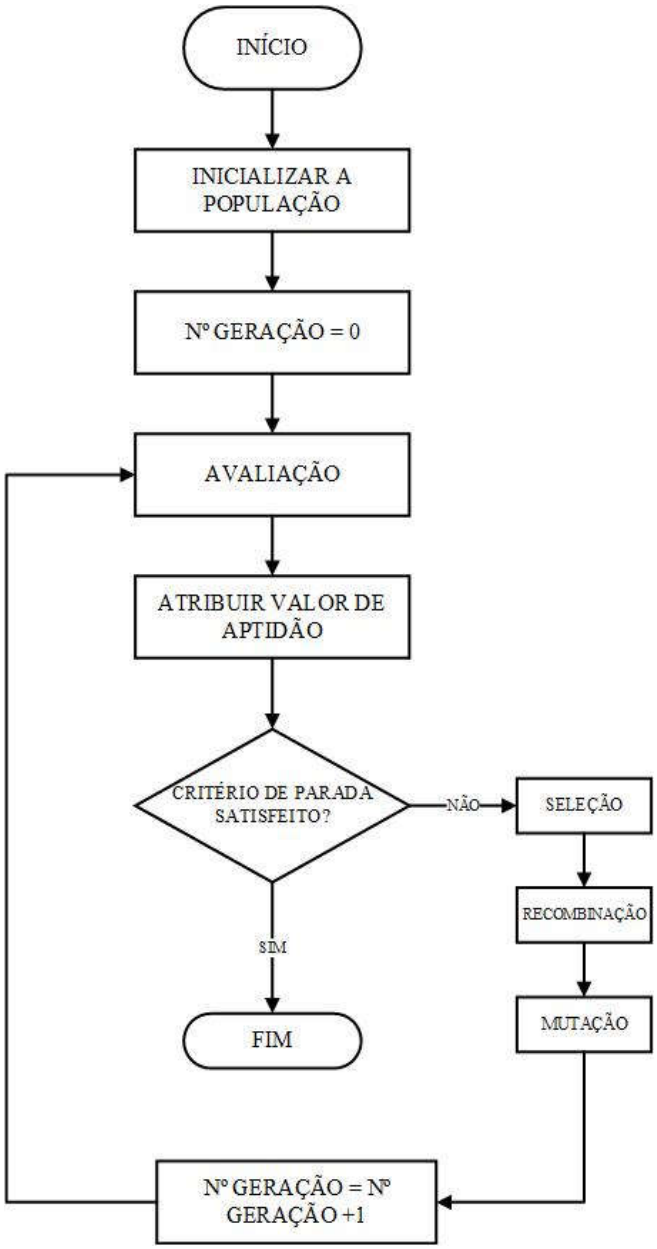
O mecanismo de mutação utilizado consiste em modificar um gene (uma porta reversível ternária) de um determinado indivíduo (circuito reversível ternário). A mutação pode ser definida em algumas etapas:

i. Define-se uma taxa de mutação T_M;
ii. Gera-se um número aleatório μ no intervalo 0 - 1;
iii. Se μ < T_M a mutação é efetuada. Senão, o indivíduo permanece igual;
iv. Seleciona-se aleatoriamente um gene do indivíduo filho. O gene escolhido é substituído.

6. Após a aplicação dos operadores genéticos de recombinação e mutação, avalia-se a os novos indivíduos modificados pelos operadores. Se os indivíduos gerados forem de melhor qualidade serão substituídos pelos indivíduos originais, formando uma nova população. A nova população torna-se a população corrente. Senão os indivíduos originais permanecem como população corrente. Deve-se retornar para a etapa 2.

Na Figura 26, apresenta-se a estrutura básica de um Algoritmo Genético Simples.

Figura 26 – Fluxograma genérico do Algoritmo Genético.



Fonte: Próprio Autor

5.2 ALGORITMO GENÉTICO SIMPLES II

A maioria das contribuições encontradas na literatura especializada consiste em empregar Algoritmos Genéticos para otimizar alguns aspectos nos projetos de circuitos

reversíveis (DRECHSLER, FINDER, WILLE; 2011; MOHAMMADI, 2012; DATTA et al. 2013;) ou são aplicados a circuitos reversíveis aritméticos (BILOSHYTSKYI; DEIBUK, 2015; KHAN;PERKOWSKI, 2007; KHAN, 2008). Dentre as combinações híbridas de algoritmos evolutivos e busca exaustiva, uma proposta originária pode ser encontrada em (LUKAC; PERKOWSKI, 2005). Recentemente, em Wang (2015) um algoritmo genético com cromossomos de comprimento variável foi combinado com um banco de dados codificado de forma ideal, semelhante ao de Golubitsky, Falconer e Maslov (2010) para obter muitos circuitos de 4 qubits. A única abordagem evolutiva conhecida para o projeto de circuitos reversíveis sem depender de um banco de dados de circuitos predefinidos, como em Wang (2015) e Golubitsky, Falconer e Maslov (2010), isto é, a partir de uma solução vazia, foi introduzida por Hadjam e Moraga (2014). Todas as contribuições mencionadas não usam algoritmos genéticos para otimizar tanto o custo quanto a quantidade de portas. O emprego de diversos critérios de otimização durante o processo de síntese pode convergir em uma variedade de novas soluções importantes.

Neste contexto, propõe-se uma segunda metodologia com base no Algoritmo Genético Simples para a síntese de circuitos reversíveis com a finalidade de otimizar tanto o custo quanto a quantidade de portas, sem depender de um banco de dados de circuitos predefinidos.

5.2.1 Descrição do algoritmo genético simples II (AGS-II)

O AGS-II para a síntese de circuitos reversíveis ternários é uma versão aprimorada do AG Simples proposto na seção anterior 5.1, em que as etapas de codificação do indivíduo, geração da população inicial, seleção, recombinação e mutação são empregadas de modo igual. A variação é em relação aos parâmetros de otimização. No processo de busca do AG Simples, utilizou-se a quantidade de portas como critério único de otimização. No AGS-II proposto, além da quantidade de portas, o custo total dos circuitos foram considerados no processo de busca pela solução. Portanto, dois componentes de aptidão são considerados:

I. Distância *Hamming*;
II. Custo Total.

Ressalta-se que na seção 5.1.3, a metodologia para obter o valor da Distância *Hamming* foi descrita. Para comparar a complexidade das realizações, as mesmas métricas empregadas ao MMD *plus* são consideradas. As portas simples são atribuídas a um custo igual a 1,

controladas por 2 portas a um custo igual a 2 e portas controladas por 0 ou 1 atribuídas a um custo igual a 4. Como já mencionado, as portas controladas por 0 ou 1 possuem maior custo devido às duas portas simples auxiliares necessárias na realização do circuito.

Atribuem-se as variáveis Q_{AGS-II} e Q_{MMD} a quantidade de portas obtidas com as realizações do AGS-II e do MMD *plus*, bem como as variáveis C_{AGS-II} e C_{MMD} o custo total das realizações determinadas pelo AGS-II e MMD *plus*. Assim sendo, consideram-se como soluções os indivíduos que satisfazem os seguintes critérios:

1. $Q_{AGS-II} < Q_{MMD}$ e $C_{AGS-II} < C_{MMD}$. Quando o AGS-II é superior que MMD *plus* em relação à quantidade de portas e custo total;
2. $Q_{AGS-II} < Q_{MMD}$ e $C_{AGS-II} \geq C_{MMD}$. Quando o AGS-II é superior que MMD *plus* somente em relação à quantidade de portas;

3. $Q_{Q_{AGS-II}} == Q_{MMD}$ e $C_{AGS-II} < C_{MMD}$. Quando o AGS-II é superior que o MMD *plus* somente em relação ao custo.

É importante destacar que, as abordagens Randômica e Menos_Um, propostas na seção 5.1.1.1, para determinar o tamanho dos indivíduos sempre geram circuitos com menor quantidade de portas reversíveis ternárias do que os gerados por MMD *plus.* Entretanto, no AGS-II a abordagem Randômica é alterada, gerando circuitos com menor ou igual quantidade de portas que o algoritmo MMD *plus.*

6 RESULTADOS EXPERIMENTAIS

Neste capítulo são apresentados os resultados obtidos por meio dos testes realizados com a implementação computacional do método de síntese de circuitos reversíveis ternários MMD *plus* e das metodologias propostas:

I. MMD *plus* com a ordenação dos vetores de entrada por distância *Hamming;*
II. MMD *plus* com a ordenação dos vetores de entrada por peso *Hamming;*
III. MMD *plus* aplicado aos modos *forward*, *backward*, *top-down* e *bottom-up;*
IV. MMD *plus* aplicado a ciclos de permutação de ordem natural;
V. MMD *plus* aplicado a transposições;
VI. MMD *plus* aplicado ciclos de tamanho 3;
VII. Síntese de circuitos Reversíveis Ternários com base no Algoritmo Genético Simples I;
VIII. Síntese de Circuitos Reversíveis Ternários com base no Algoritmo Genético Simples II.

As metodologias citadas acima foram enumeradas de I a VIII com a finalidade de simplificar a apresentação dos resultados. Para a implementação computacional de tais metodologias foi utilizada a linguagem de Programação Python 2.7. A implementação foi executada em um processador Intel® Core ™ i5-4440 com 8 GB de memória principal, operando a 3.10 GHz sob o sistema operacional Windows 7 de 64 bits. Como referência utilizou-se o conjunto de todas as possíveis funções reversíveis ternárias de 2 variáveis, assim como foi realizado em MMD_T (MILLER; MASLOV; DUECK, 2006). Uma vez que existem 9 possibilidades de atribuições para os vetores que representam uma função reversível ternária de 2 variáveis, estes vetores podem ser permutados em 9! = 362880 ordens distintas. As 362880 permutações foram obtidas com o algoritmo Heap (HEAP, 1963) disponível no software Scilab ™. O MMD *plus* foi aplicado pela primeira vez para executar como referência todas as funções mencionadas, na sua forma padrão, isto é, com os vetores de entrada ordenados lexicograficamente, realizando uma busca *backward* e *top-down*. O tempo médio de execução do algoritmo e verificação dos circuitos foi de 4.2 milissegundos por função reversível ternária.

6.1 EXPERIMENTOS REALIZADOS COM AS METODOLOGIAS I E II

Para uma comparação equilibrada, a mesma implementação do algoritmo de síntese MMD *plus* foi utilizada como base para a implementação de todas as metodologias propostas, que são variações do algoritmo originário MMD_T. Assim como, utilizou-se a mesma abordagem descrita na seção 3.3.2 do Capítulo 3 para calcular o custo total. As únicas variáveis independentes consistem na ordenação dos vetores de entrada, por distância e peso *Hamming*, e nos vetores de entrada ou saída que definem o início do processo de busca do algoritmo, sendo as buscas orientadas em modos *forward*, *backward*, *top-down* e *bottom-up.*

Posteriormente a aplicação do algoritmo MMD *plus* na forma padrão ao *benchmark* com as 360, 880 mil funções reversíveis ternárias, realiza-se a aplicação do algoritmo, em que os vetores de entrada são ordenados por distância e por peso *Hamming,* divergindo da forma de ordenação padrão (lexicográfica) do MMD *plus*. Na Tabela 18 estão sumarizados os resultados, apresentando a distribuição da quantidade de funções realizadas de acordo com a quantidade de portas necessárias para a determinação do circuito. As portas são do tipo simples ou controladas, independentemente do valor do trit de controle. É possível constatar que o MMD *plus* na sua forma padrão gerou circuitos reversíveis ternários com no máximo 12 portas, enquanto que a Metodologia I, em que os vetores de entrada foram ordenados por distância *Hamming*, obteve-se para 27 funções reversíveis ternárias circuitos com até 13 portas. Bem como a Metodologia II, cujos vetores de entrada são ordenados de acordo com o peso *Hamming*, foram gerados circuitos com até 14 portas reversíveis ternárias.

Tabela 18 – Quantidade de portas reversíveis ternárias e das funções realizadas.

Realizações com o MMD *plus*			
Nº de portas	**Padrão**	**Metodologia I**	**Metodologia II**
	Nº de funções	**Nº de funções**	**Nº de funções**
1	40	40	40
2	690	690	692
3	6137	6137	6160
4	28383	28538	28232
5	64508	65599	61716
6	91788	92348	84253
7	84262	81811	77173
8	54457	52468	53590
9	23898	24705	29236
10	7187	8231	13589
11	1358	1973	5369
12	171	312	2231
13	---	27	562
14	---	---	36

Fonte: Próprio Autor

Sabe-se que as portas controladas têm diferentes custos, dependendo do valor do trit de controle. Quando o trit de controle é 2, o custo atribuído é igual a 2, e quando o trit de controle é 0 ou 1, atribui-se o custo igual a 4. Portanto, os circuitos reversíveis ternários com a mesma quantidade de portas contêm, comumente, um custo total diferente. Esta situação é comprovada por meio dos testes apresentados na Tabela 19. O método MMD *plus*, por exemplo, é superior a Metodologia I em 12% do *benchmark*, com relação à quantidade de portas nas realizações. Por outro lado, em relação ao custo das realizações, apresenta superioridade de 21% sobre a Metodologia I.

Tabela 19 – Quantidade de funções realizadas em uma comparação experimental do MMD plus versus Metodologias I e II.

Condição	Com relação ao nº de portas	Com relação ao Custo Total
MMD *plus* é superior que Metodologia I	46544 funções	78733 funções
Metodologia I é superior que MMD *plus*	46291 funções	51049 funções
Metodologia I e MMD *plus* apresentam resultados iguais	270044 funções	233097 funções
MMD plus é superior que Metodologia II	68604 funções	80696 funções
Metodologia II é superior que MMD *plus*	33548 funções	44903 funções
Metodologia II e MMD *plus* apresentam resultados iguais	260727 funções	237280 funções

Fonte: Próprio Autor

6.2 EXPERIMENTOS REALIZADOS COM A METODOLOGIA III

Como mencionado, Miller, Maslov e Dueck (2006), autores de MMD, mostraram que os modos backward, forward e bidirecional podem levar a circuitos com complexidades diferentes. Isto se deve ao fato de que o algoritmo é ganancioso e, a cada passo, procura por um ótimo local. A sequência de visualizações locais não é necessariamente a mesma em um modo backward, em que o processo de síntese é iniciado a partir dos vetores de saída, e forward, em que é iniciado a partir vetores de entrada. Foram realizados testes com a finalidade de comprovar se o mesmo ocorre quando algoritmo é processado no modo top-down, isto é, de cima para baixo ou no modo bottom-up, de baixo para cima. Utilizou-se o benchmark com as 362880 funções reversíveis ternárias de 2 variáveis. Os resultados são expostos na Tabela 20.

Em relação ao custo total dos circuitos reversíveis ternários obtidos, aproximadamente 70% são superiores, possuindo menor custo, com o emprego do modo *Top-down,* tanto na orientação de busca nos modos *backward* e *forward*. E em 21% do *benchmark*, os circuitos gerados com o emprego do modo *Bottom-up* mostraram-se superiores, também em relação aos custos e para os modos *backward* e *forward.* Portanto, somente em aproximadamente 9% os

resultados foram semelhantes, comprovando que as diferentes orientações no processo de síntese geram circuitos com quantidade de portas e custos diferentes.

Tabela 20 - Quantidade de funções realizadas em uma comparação experimental do MMD *plus* versus Metodologias I e II.

	Condição	**Com relação ao nº de portas**	**Com relação ao Custo Total**
MMD *plus* - *Backward*	Modo Top-down é superior que Bottom-up	131998 funções	252141 funções
	Modo Bottom-up é superior que Top-down	133752 funções	78162 funções
	Ambos os modos apresentam resultados iguais	97129 funções	32576 funções
MMD *plus* - *Forward*	Modo Top-down é superior que Bottom-up	131998 funções	252141 funções
	Modo Bottom-up é superior que Top-down	133752 funções	78162 funções
	Ambos os modos apresentam resultados iguais	97129 funções	32576 funções

Fonte: Próprio Autor

Em relação ao modo bidirecional, que é uma combinação dos modos backward e forward, foram obtidas melhoras significativas em relação à quantidade de portas. As realizações com os métodos MMD plus na versão padrão (backward e top-down) e bidirecional foram comparadas na Tabela 21.

.

Tabela 21 - Realizações com os métodos MMD plus na versão padrão (backward e top-down) e bidirecional.

Realizações com o MMD *plus*		
Nº de portas	**Padrão**	**Bidirecional**
	Nº de funções	**Nº de funções**
1	40	40
2	690	732
3	6137	7646
4	28383	41561
5	64508	94769
6	91788	114724
7	84262	72017
8	54457	26072
9	23898	4812
10	7187	480
11	1358	26
12	171	---
13	---	---
14	---	---

Fonte: Próprio Autor

O MMD *plus* na versão bidirecional determinou circuitos com no máximo 11 portas, enquanto que o MMD *plus* na forma padrão para o mesmo *benchmark* produziu circuitos com até 12 portas, para 171 realizações.

Ademais, as realizações obtidas com o método MMD *plus* também foram comparadas com o método original MMD_T, ambos no modo bidirecional. O MMD *plus,* como mencionado, determinou circuitos com no máximo 11 portas reversíveis ternárias do tipo MS, especificamente para 26 realizações, enquanto que o MMD_T gerou circuitos com até 12 portas, para 84 realizações. Dessa forma, é possível constatar que o emprego das portas do tipo MS na abordagem estendida MMD *plus* foi favorável à redução da quantidade de portas dos circuitos gerados. Ressalta-se que em MMD_T foram utilizadas as portas reversíveis ternárias propostas por De Vos, Raa e Storme (2002), que são uma adequação das portas reversíveis binárias NOT, Toffoli e Feynman (TOFFOLI, 1980; FEYNMAN,1986).

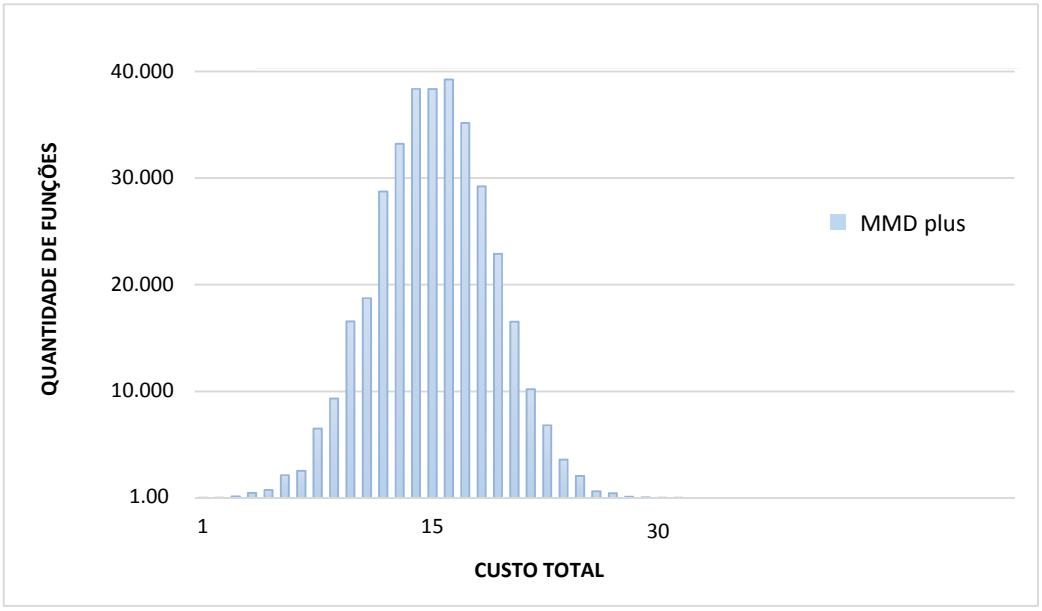
6. 3 EXPERIMENTOS REALIZADOS COM AS METODOLOGIAS IV, V E V

O mesmo algoritmo de síntese do MMD *plus* empregado ao *benchmark* contendo todas as 362880 possíveis funções reversíveis ternárias de 2 variáveis, foi empregado às funções decompostas na forma de ciclos de permutação de ordem natural. Preliminarmente, os ciclos foram determinados por meio de um algoritmo implementado também em Python 2.7 e o MMD *plus* foi aplicado na realização dos ciclos. Os custos das realizações com e sem a decomposição em ciclos ordem natural foram calculados. O custo médio de realização das funções reversíveis ternárias com a aplicação direta do MMD *plus* foi de 15 e o custo médio de realizações com base na decomposição em ciclos de ordem natural, metodologia IV, foi de 19. A distribuição é mostrada na Figura 27.

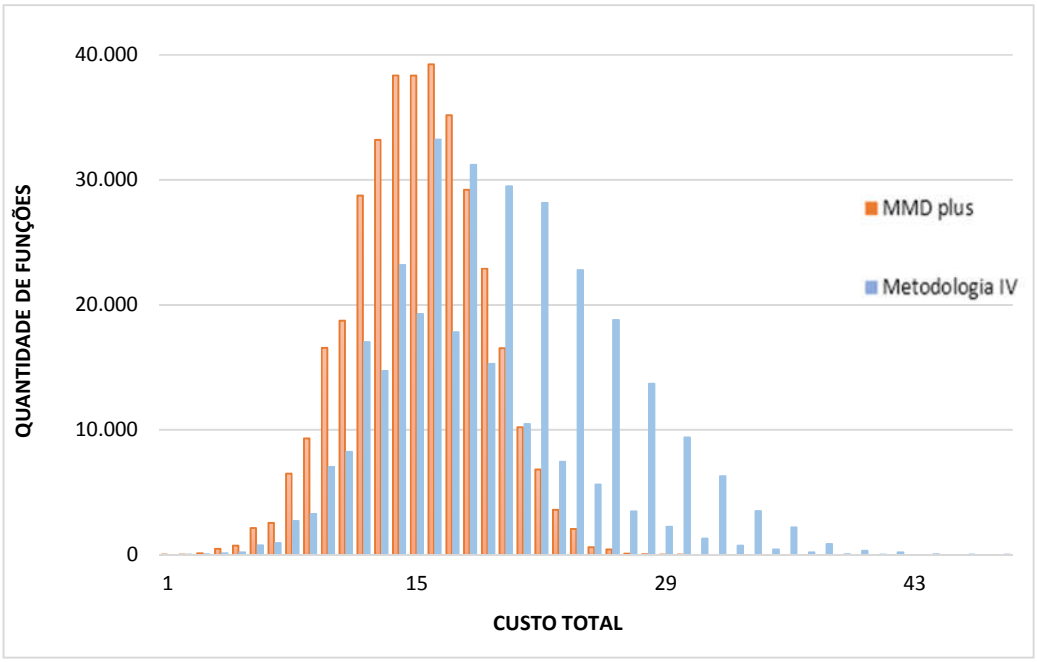
Pode ser observado, na Figura 27 (b), que duas distribuições entrelaçadas aparecem. Uma superior, quando o custo de realização é par, e outra inferior quando o custo é ímpar. Tal característica deve-se ao fato de que o custo será ímpar quando uma porta simples for utilizada em uma extremidade do circuito, ressaltando que todas as portas controladas possuem custo par, de 2 ou de 4. As portas simples não podem ser dispostas em alguma posição intermediária dos circuitos, visto que quando utilizada não preserva o prefixo já adaptado entre as especificações de entrada e saída. Torna-se notório que a maioria dos ciclos são realizados usando portas reversíveis ternárias controladas, enquanto que as realizações do MMD *plus* as funções não decompostas, frequentemente, usam portas simples em uma extremidade. O circuito da Figura 17 da seção 4.3.4 constitui um exemplo que demonstra tal situação, pois quando a função f_2 foi realizada com o MMD *plus*, obteve-se um circuito com o custo total de 21 e 8 portas, sendo 7 portas controladas e uma porta simples. Quando mesma função foi realizada na forma de ciclos de permutação de ordem natural obteve-se um circuito com custo total de 14 e 5 portas, cujas portas são todas controladas. A distribuição do número de funções em relação à quantidade de portas necessárias a síntese em ambos os algoritmos é mostrada na Tabela 22.

Demais resultados estão sumarizados na Tabela 23.

Figura 27 – Distribuição da quantidade de funções de acordo com o custo total do circuito sintetizado. (a) com MMD *plus* (b) com a Metodologia IV



(a)



(b)

Fonte: Próprio Autor

Tabela 22 –Quantidade de portas reversíveis ternárias e das funções realizadas.

Nº de portas	**MMD *plus***	**Metodologia IV**
	Nº de funções	**Nº de funções**
1	40	30
2	732	517
3	7646	5242
4	41561	27036
5	94769	59332
6	114724	73751
7	72017	67854
8	26072	53097
9	4812	34722
10	480	21962
11	26	10930
12	---	5572
13	**---**	1816
14	**---**	756
15	**---**	214
16	**---**	32
17	**---**	16

Fonte: Próprio Autor

Tabela 23 - Quantidade de funções realizadas em uma comparação experimental do MMD *plus* versus Metodologias IV.

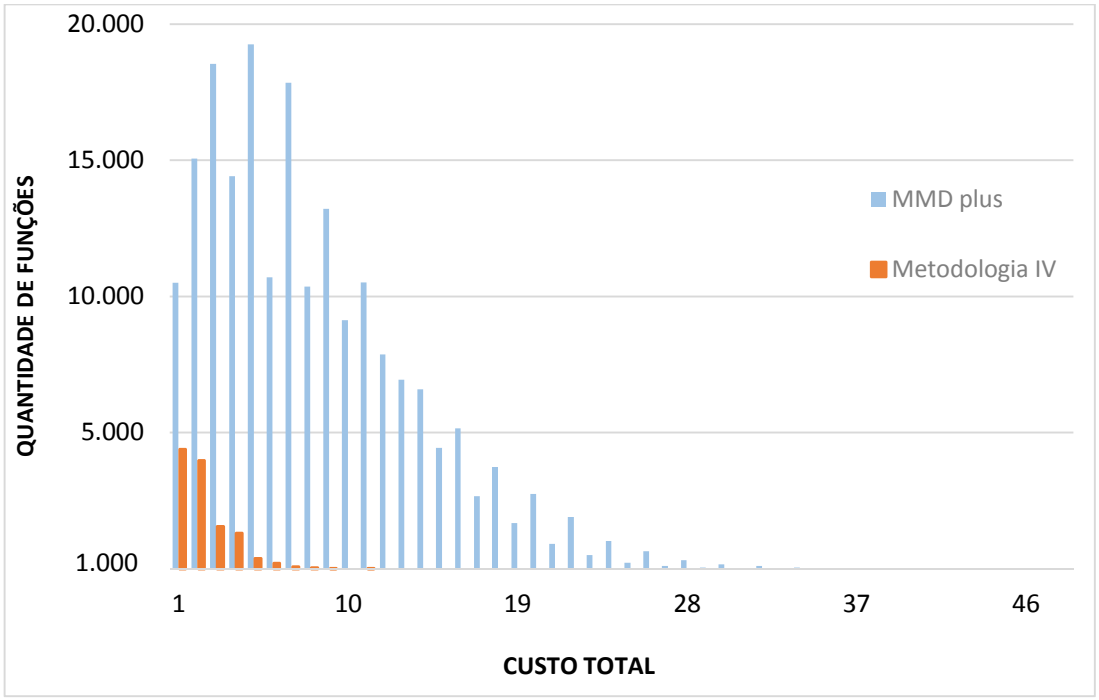
Condição	**Com relação ao nº de portas**	**Com relação ao Custo Total**
MMD *plus* é superior que Metodologia IV	164483 funções	197392 funções
Metodologia IV é superior que MMD *plus*	15261 funções	11919 funções
Metodologia IV e MMD *plus* apresentam resultados iguais	183135 funções	153568 funções

Fonte: Próprio Autor

Pode ser visto que aproximadamente 54,39% das funções realizadas com MMD *plus* têm menor custo total do que as realizações obtidas com base nos ciclos de permutação de ordem natural. Ressalta-se que as portas controladas têm um custo diferente dependendo do valor do trit de controle. Portanto, os circuitos com o mesmo número de portas podem comumente apresentar custos diferentes.

A faixa de melhoria de custo do MMD *plus* em relação à Metodologia IV cresce de 1 para 46, com uma média ponderada (abscissa do centro de gravidade da distribuição) igual a 8,19. Apenas em aproximadamente 3,3% dos casos a realização baseada em ciclos alcançou um custo menor do que MMD *plus*, com uma faixa de melhoria de custo entre 1 e 11, com uma média ponderada de 2,2. Caso contrário, ambos os métodos alcançam o mesmo custo. A distribuição do número de funções em relação às melhorias de custo é mostrada na Figura 28. Em cor azul, quando o custo de MMD *plus* é menor que o custo de realizações da Metodologia IV. Na cor alaranjada, quando o custo da Metodologia IV é menor que o custo de realizações do MMD *plus*.

Figura 28 – Distribuição da quantidade de funções de acordo com a melhoria de custo total apresentada por MMD *plus* e Metodologia IV.

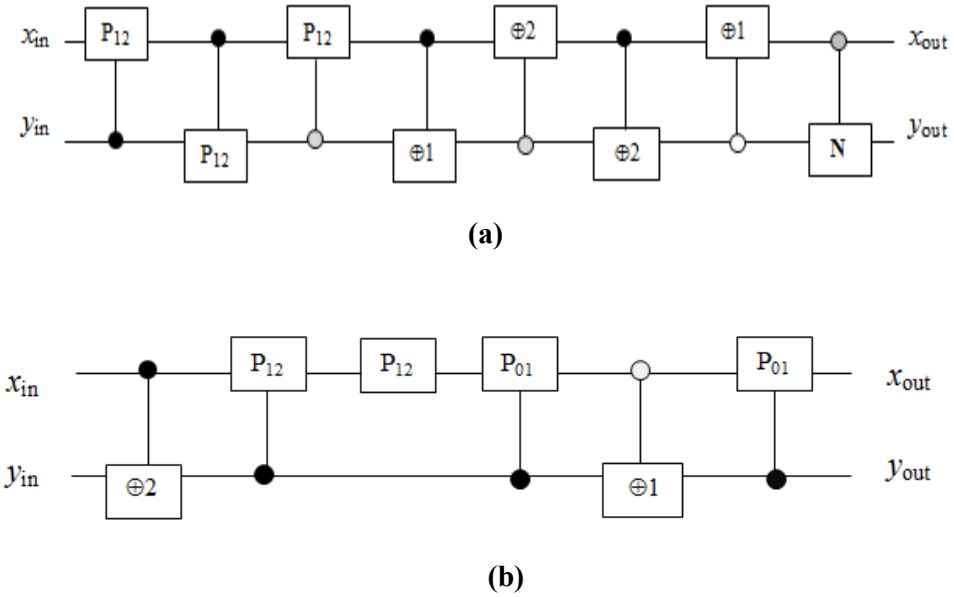


Fonte: Próprio Autor

Não esperou que, em uma grande quantidade de funções, a realização do MMD *plus* a funções decompostas em ciclos de ordem natural fosse menor que a realização direta, pois os ciclos de permutação têm restrições especiais, como, por exemplo, quando um determinado ciclo é aplicado aos vetores de entrada para obter a sua representação na forma de tabela de valores, faz-se necessário preservar todos os elementos que não compõem o ciclo durante o processo de síntese. Um resultado inesperado é que, em um pequeno número de funções, especificamente em 3,3% de todo o *benchmark*, a realização baseada em ciclos tem um custo menor do que a realização direta de MMD *plus*, posto que os ciclos também são realizados com MMD *plus*. Destaca-se, no entanto, que o MMD *plus* é um algoritmo guloso, que otimiza sempre a iteração seguinte, sem considerar a influência que a atual iteração pode ter no restante de um processo de síntese. Se uma determinada função reversível ternária contiver dois ciclos, então, MMD *plus* engloba três problemas diferentes. A realização da função original e a dos dois ciclos, e pode ocorrer que em um ou ambos os ciclos sejam menos complexos de realizar (e combinam-se como uma cascata) do que a realização de toda a função. Um caso extremo é mostrado abaixo no Exemplo 1, em que uma determinada função reversível ternária, que é por definição uma permutação, é decomposta em dois ciclos. Os circuitos gerados, por meio da aplicação de MMD *plus* aos ciclos, diferem em duas portas e em um custo igual a 11, a favor dos ciclos, como mostrado na Figura 29. Esta é, entretanto, a única função em todo o *benchmark*.

Exemplo 1: Dada uma função reversível ternária f_3 = (12, 00, 02, 20, 21, 01, 11, 22, 10), a decomposição em ciclos de permutação de ordem natural correspondente é (12, 01, 00) (20, 11, 21, 22, 10). Emprega-se o processo de síntese do MMD *plus* a função f_3 como uma única permutação, bem como descomposta em uma cascata de ciclos. Os circuitos obtidos são apresentados na Figura 29 (a) e (b), respectivamente. O custo total da realização direta do MMD *plus* a função f_3 foi de 24, e da realização a cascata de ciclos foi de 11.

Figura 29 – (a) Realização do MMD plus com custo igual a 24 e em (b) realização aos ciclos de permutação.



Fonte: Próprio Autor

Em Shende et al. (2003) foi introduzida a síntese de circuitos reversíveis binários em transposições, que são ciclos de permutação com 2 elementos. Sasanian et al. (2009) provaram que usando decomposições em ciclos de tamanho 3, em muitos casos, levaria a realizações com menos portas reversíveis. Analogamente, uma análise semelhante no contexto ternário, utilizando como *benchmark* todo o conjunto de 362880 funções reversíveis ternárias de duas variáveis, foi realizada. Os resultados estão sumarizados na Tabela 24. Pode-se observar que em 96% do *benchmark* os circuitos obtidos com a Metodologia IV conduzem a realizações de menor custo do que decompondo os ciclos em transposições, Metodologia V. Além disso, em 69,5% dos circuitos do *benchmark*, as realizações com ciclos de ordem natural têm um custo menor do que com os ciclos com 3 elementos. Esses resultados são semelhantes aos de Sasanian et al. (2009) no contexto binário e permitem afirmar que a decomposição adicional de ciclos em transposições ou ciclos com 3 elementos não seria eficiente para a síntese de circuitos reversíveis ternários. Somente para poucas funções, em uma parte do *benchmark* tais Metodologias levariam a melhorias de custo com relação ao custo de realização direta de MMD *plus*, isto é, a funções reversíveis ternárias sem decomposições.

Tabela 24 - Quantidade de funções realizadas em uma comparação experimental das Metodologias IV, V e VI.

Condição	Com relação ao nº de portas	Com relação ao Custo Total
Metodologia IV é superior que Metodologia V	33910 funções	348,004 funções
Metodologia V é superior que Metodologia IV	11884 funções	7079 funções
Metodologia IV e V apresentam resultados iguais	13085 funções	7,796 funções
Metodologia IV é superior que Metodologia VI	223473 funções	252204 funções
Metodologia VI é superior que Metodologia IV	92792funções	86985 funções
Metodologia IV e VI apresentam resultados iguais	46614 funções	23690 funções

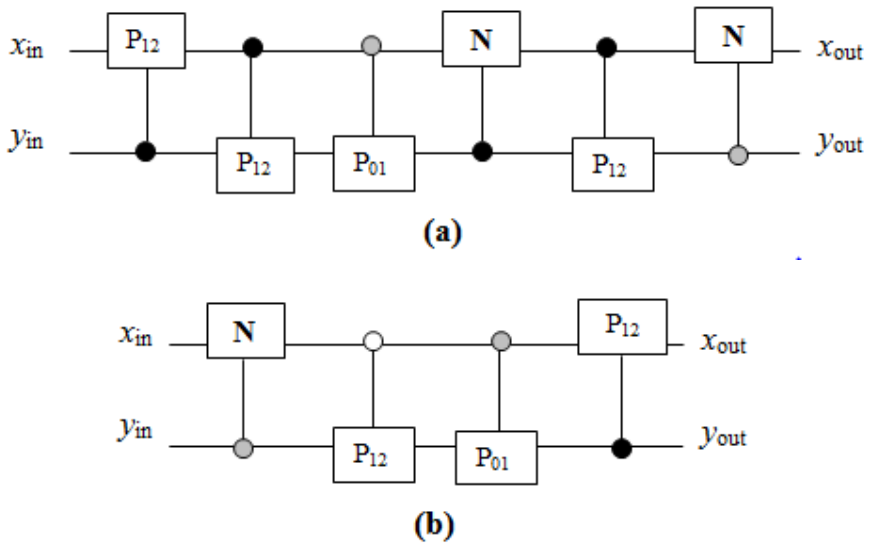
Fonte: Próprio Autor

É válido ressaltar que existem algumas transposições com baixo custo. Em particular as transposições com a estrutura (v, ($v \oplus w$)) ou (($v \oplus w$), v) onde $v \in \{0,1, 2\}$, $w \in \{1, 2\}$ e $\oplus$ denota a soma *módulo* 3. Se for dada uma permutação global e a decomposição leva a transposições de baixo custo, então esta decomposição apresentará uma realização de menor custo do que a com base na realização do MMD *plus*. Esta situação acontece em apenas 1,9% das funções de *benchmark*. Ressaltando que o algoritmo MMD *plus* é ganancioso e sempre tenta otimizar etapa de iteração predominante, seguindo uma orientação *forward* para adaptar as entradas para as saídas especificada (ou vice-versa).

Considerando a função reversível ternária f_4 = (00, 21, 01, 11, 10, 22, 20, 02, 12), como exemplo, realiza-se a decomposição em ciclos de permutação de ordem natural (01, 21, 02) (10, 11) (12, 22) (6) e posteriormente em transposições (01, 21) (01, 02) (10, 11) (12, 22). Com a realização direta do MMD *plus* foi gerado um circuito com 6 portas reversíveis ternárias e com custo total de 16, como pode ser visualizado na Figura (a).

Cada uma das transposições é realizada com uma única porta. O circuito obtido a partir das transposições é ilustrado na Figura (b). As transposições (01, 21) (01, 02) (10, 11) (12, 22) tem um custo igual a 4, 4, 4 e 2, respectivamente, somando um custo total igual a 14.

Figura 30 - (a) Realização do MMD *plus* com custo igual a 16 e em (b) realização como um produto de transposições, com um custo igual a 14.



Fonte: Próprio Autor

6.4 EXPERIMENTOS REALIZADOS COM OS ALGORITMOS GENÉTICOS SIMPLES I (AGS) E II (AGS-II) PROPOSTOS

Apresentou-se um método com base no AGS para a síntese de circuitos reversíveis ternários utilizando-se das portas reversíveis ternárias do tipo MS, definidas no Capítulo 3. Nesta seção, são mostrados e discutidos os resultados obtidos por meio dos experimentos realizados com um benchmark contendo mil funções reversíveis ternárias. Tais funções foram selecionadas aleatoriamente a partir do benchmark completo que contém todas as 362880 possíveis funções de duas variáveis.

Para comparar a eficiência do AGS e do AGS-II propostos, utilizam-se como referência os resultados gerados com o MMD *plus* na versão padrão. Dois critérios são considerados para mensurar a qualidade dos resultados obtidos:

I. A quantidade de portas dos circuitos, independente do valor de controle. É válido ressaltar que o tamanho do indivíduo equivale à quantidade de portas;

II. O custo total dos circuitos. O custo total é definido pelo somatório do custo individual de cada porta reversível ternária que compõe o circuito.

Para a implementação computacional do AGS e do AGS-II foi utilizada a linguagem de Programação Python 2.7. As implementações foram executadas em um processador Intel® Core ™ i5-4440 com 8 GB de memória principal, operando a 3.10 GHz sob o sistema operacional Windows 7 de 64 bits. Os experimentos foram realizados combinando diferentes parâmetros do AGS, com a finalidade de analisar os possíveis impactos na qualidade dos circuitos gerados. Foram determinadas populações aleatórias com 50, 100 e 200 indivíduos, empregando uma taxa de recombinação de 50% e de mutação de 0,4 em todos os experimentos. No processo de seleção dos indivíduos pais de melhor aptidão, duas metodologias foram utilizadas, a de seleção por Elite e por Proximidade de Inteiros. Além disso, foram empregadas, para definir o tamanho do indivíduo na geração da população inicial, as abordagens propostas Menos-Um e Randômica.

Tabela 25 – Quantidade de circuitos superiores gerados pelo AGS, utilizando-se como parâmetro de comparação o algoritmo MMD *plus* na versão padrão.

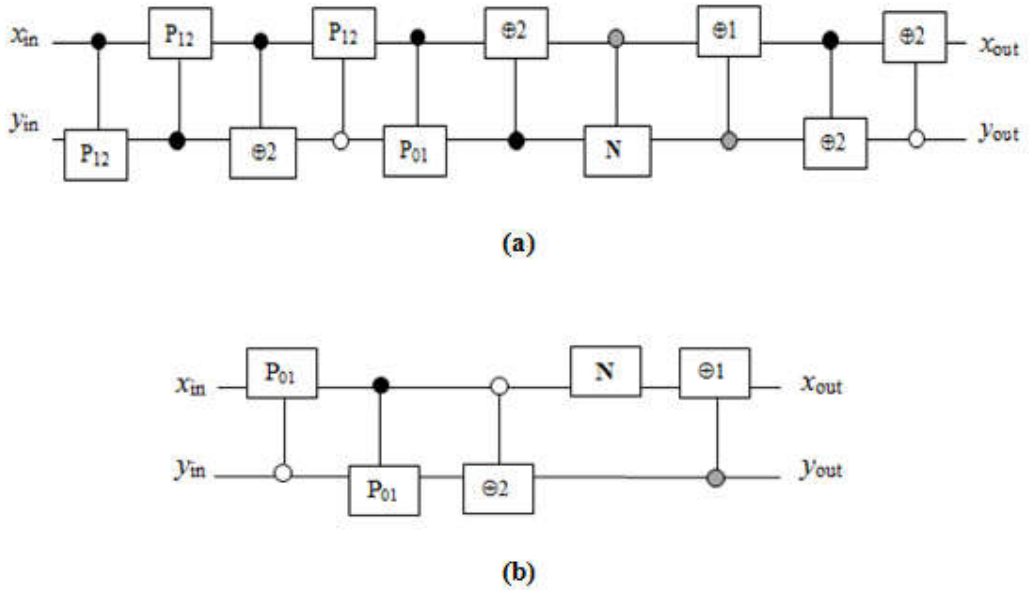
Seleção por Elite				
Tamanho dos indivíduos	**Quantidade de evoluções**	**Tamanho da População**	**Nº de circuitos superiores**	**Nº de portas reduzidas**
RANDÔMICA	**20**	50	118	186
		100	245	413
		200	400	677
	40	50	201	307
		100	311	495
		200	398	664

Fonte: Próprio Autor

Na Tabela 25, mostra-se experimentos realizados sob o *benchmark* reduzido com mil funções reversíveis ternárias, em que a quantidade de portas dos circuitos gerados pelo AGS proposto foi comparada com a quantidade gerada pelo algoritmo MMD *plus* padrão. É possível constatar que, o AGS configurado a uma população de 200 indivíduos, realizando a seleção por Elite e tamanho dos indivíduos determinados randomicamente, foram obtidos em 40 % do benchmark circuitos superiores, isto é, com menor quantidade de portas do que os gerados pelo MMD *plus*. Este aspecto mostra que o MMD *plus* não gera soluções ótimas garantidas.

Ademais, é possível notar que além de apresentar superioridade em 40% do *benchmark*, a redução da quantidade de portas foi significativa, pois para algumas funções, especificamente 27%, foram reduzidas mais que uma porta por circuito. Em particular, para a função reversível ternária F = (10, 22, 20, 11, 01, 12, 21, 00,02) o AGS gerou um circuito com 5 portas e com custo total igual a 14, ao passo que o MMD plus para a mesma função determinou um circuito com 10 portas e com custo total de 28. É de suma importância ressaltar que o AGS, para a configuração mencionada, requer em média 16 segundos para a determinação do circuito a partir da especificação dada, enquanto que o MMD *plus* demanda em média em 4,2 milissegundos, ou seja, o AGS pode realizar circuitos ótimos ou próximos do ótimo, porém demanda maior tempo de realização. Os circuitos gerados com o AGS proposto e o MMD plus são apresentados na Figura 31-(a) e (b), respectivamente.

Figura 31- Realizações para a função reversível ternária *f* = (10, 22, 20, 11, 01, 12, 21, 00,02). (a) com MMD *plus* e (b) com AGS.

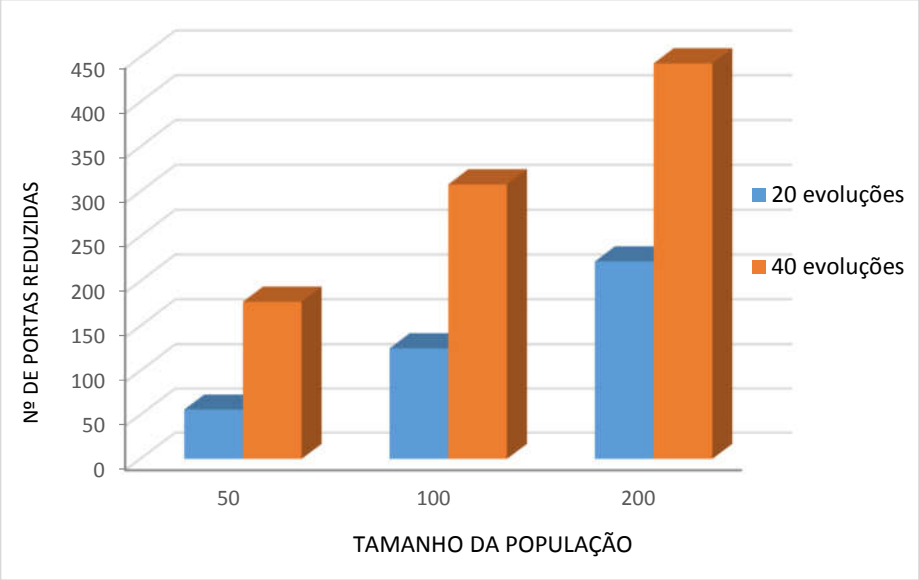


Fonte: Próprio Autor

Na Figura 32, um gráfico é apresentado com a finalidade de mostrar que o tamanho da população e a quantidade de evoluções influenciaram diretamente na quantidade de circuitos superiores determinados em relação ao MMD *plus*. Os resultados mostrados foram obtidos para populações de 50, 100 e 200 indivíduos, com 20 e 40 evoluções, em que a abordagem de seleção utilizada foi por Proximidade de Inteiros. O tamanho dos indivíduos foi configurado por meio

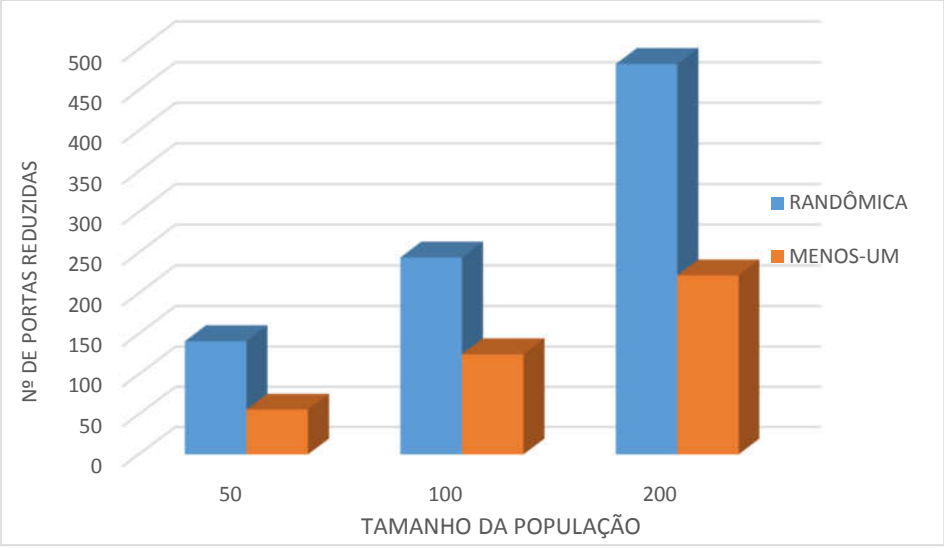
da metodologia Menos-Um. Pode-se notar que à medida que o tamanho da população e a quantidade de evoluções acrescem, o número de portas reduzidas nos circuitos aumenta proporcionalmente, gerando circuitos de melhor qualidade.

Figura 32 – Distribuição da quantidade de portas reduzidas em relação ao tamanho da população e o número de evoluções.



Fonte: Próprio Autor.

Figura 33 - Distribuição da quantidade de portas reduzidas sob as abordagens Randômica e Menos_Um.



Fonte: Próprio Autor.

Tabela 26 - Quantidade de funções otimizadas em uma comparação experimental do AGS proposto.

Seleção por Proximidade de Inteiros						
Tamanho dos indivíduos	**Quantidade evoluções**	**Tamanho da população**	**Tempo Total**	**Tempo médio por função (segundos)**	**Nº de circuitos superiores**	**Nº de portas reduzidas**
RANDÔMICA	**20**	50	02h40min	9	80	141
		100	04h39min	16	156	244
		200	05h06min	20	286	482
	40	50	03h26min	10	242	376
		100	05h24min	19	371	634
		200	10h02min	36	513	891
MENOS UM	**20**	50	02h07min	7	56	56
		100	04h28min	15	125	125
		200	08h33min	29	222	222
	40	50	03h00min	10	177	177
		100	05h48min	20	308	308
		200	10h35min	36	443	443

Fonte: Próprio Autor.

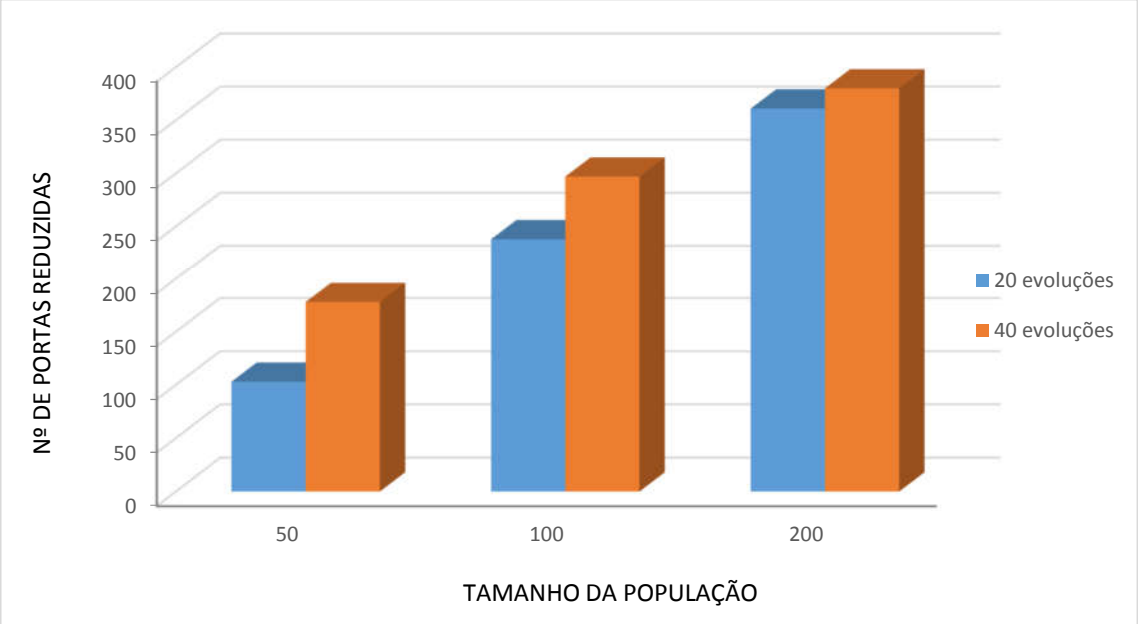
Por meio da Figura 33, mostra-se que a abordagem Randômica é significativamente superior em relação à Menos_Um, gerando para uma maior quantidade de funções, circuitos com menos portas reversíveis ternárias. Este tipo de comportamento foi esperado, uma vez que escolhendo o tamanho do indivíduo randomicamente, entre um mínimo e máximo predefinidos, a possibilidade de reduzir mais que uma porta por circuito é maior, e na abordagem Menos_Um sempre são determinados circuitos com menos uma porta em relação aos gerados por MMD plus. Uma vez que na abordagem Randômica o AGS é executado n vezes, dentre um limite mínimo e máximo definidos para cada função reversível ternária a ser sintetizada, esperava-se que o tempo de execução seria significativamente maior. Entretanto, com os experimentos realizados em diferentes configurações para o AGS, foi possível constatar que o tempo de

execução das duas abordagens para escolha do tamanho dos indivíduos são similares, como apresenta-se na Tabela 26.

Considerando os resultados obtidos com o AGS-II proposto, constatou-se um comportamento similar ao AGS, no qual o tamanho da população e a quantidade de evoluções influenciaram diretamente na quantidade de circuitos superiores determinados em relação aos de MMD *plus*. Os resultados mostrados foram encontrados a partir de populações contendo 50, 100 e 200 indivíduos, com 20 e 40 evoluções, em que a abordagem de seleção empregada foi por Elite. O tamanho dos indivíduos foi configurado por meio da abordagem Randômica. Destaca-se que o mesmo *benchmark* reduzido contendo mil funções reversíveis ternárias é utilizado em todos os experimentos, e que a população inicial é gerada aleatoriamente. Portanto, para cada vez que o algoritmo é executado tem-se uma população diferente, bem como resultados diferentes.

Pode-se notar que à medida que o tamanho da população e a quantidade de evoluções crescem a quantidade de circuitos determinados apresentando superioridade em relação aos de MMD *plus* aumenta proporcionalmente, como mostrado na Figura 34. No AGS-II, a qualidade dos circuitos é em relação à quantidade de portas e custo total.

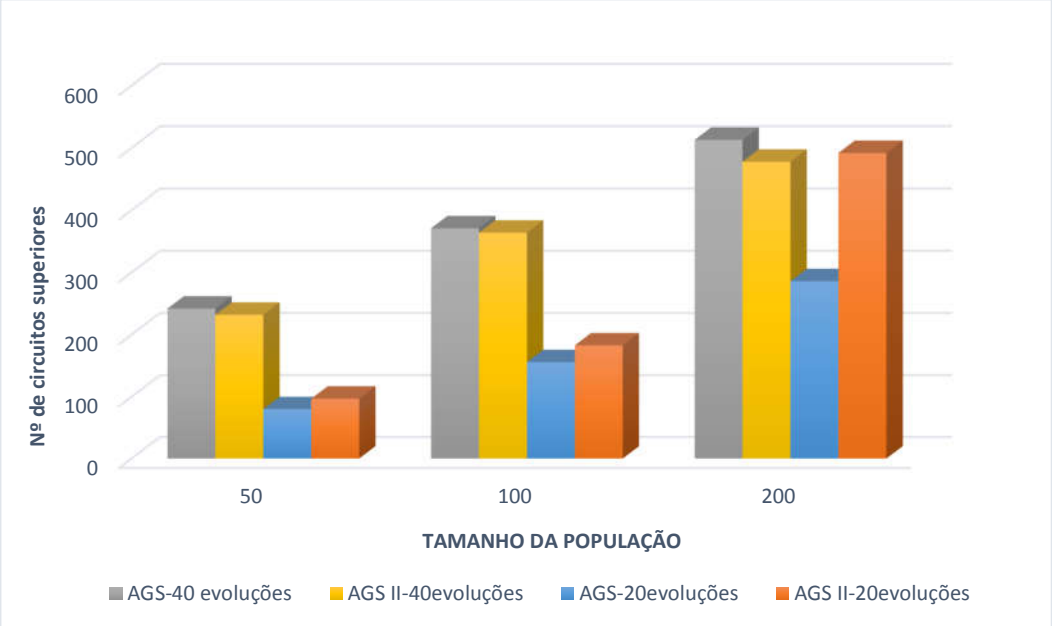
Figura 34 - Distribuição da quantidade de portas reduzidas em relação ao tamanho da população e o número de evoluções.



Fonte: Próprio Autor

Na Figura 35, são apresentados resultados comparativos entre o AGS e AGS-II. Os experimentos foram realizados com populações de 50, 100 e 200 indivíduos, cujo método de seleção foi por Proximidade de Inteiros e o tamanho dos indivíduos determinados de acordo com a abordagem Randômica. Pode-se notar que o AGS-II foi superior ao AGS sob 20 evoluções para as três variações de tamanho da população. Por outro lado, com 40 evoluções o AGS se mostrou superior.

Figura 35 – Experimentos comparativos entre AGS e AGS-II.



Fonte: Próprio Autor

Ressalta-se que no AGS-II, os indivíduos com a mesma quantidade de portas e com menor custo total que MMD *plus* fazem parte do conjunto de solução. Em virtude disso, a abordagem Menos_Um para determinação do tamanho dos indivíduos não foi empregada, uma vez que nesta abordagem sempre são gerados circuitos com menos uma porta em relação aos realizados por MMD *plus* e, dessa forma, não seria possível obter circuitos com a mesma quantidade de portas e com menor custo total que MMD *plus*.

7 OTIMIZAÇÃO PÓS-SÍNTESE PARA SÍNTESE DE CIRCUITOS REVERSÍVEIS TERNÁRIOS

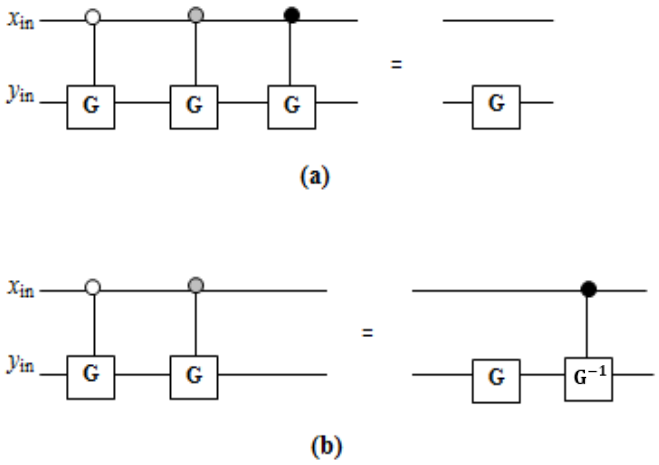
Com a realização de testes exaustivos em um grande *benchmark*, para obter resultados com uma estatística significante, como neste trabalho, o pós-processamento pode ser considerado se os padrões representativos dos circuitos puderem ser reconhecidos imediatamente à síntese como, por exemplo, quando duas portas reversíveis ternárias são vizinhas e controladas por 1 e por 0, uma vez que duas portas simples, necessárias entre os controles 0 e 1, podem ser mescladas, reduzindo assim o custo do circuito em 1.

Uma situação totalmente diferente ocorre quando se tem o problema de projetar um circuito reversível ternário para uma função reversível ternária específica. Diferentes alternativas podem ser comparadas dentro de um tempo de execução computacional razoável. No algoritmo MMD *plus,* a ordem dos elementos dos ciclos originados da decomposição é aleatória. Se, em vez disso, todas as alternativas forem testadas, uma solução ótima pode ser obtida. No caso dos ciclos, uma reordenação cíclica pode originar um menor custo de decomposição em transposições, como por exemplo, o ciclo de permutação (1 3 5 7) quando decomposto resulta nas seguintes transposições (1 3) (1 5) (1 7). Estas transposições quando realizadas com o algoritmo MMD *plus* determinam um circuito reversível ternário com custo total igual 28. Por outro lado, reordenando os elementos contidos no ciclo, obtêm-se as transposições (7 1 3 5) = (7 1) (7 3) (7 5), que determinar um circuito com custo igual a18.

Em particular, uma situação pós-processamento efetiva, associada ao modelo binário proposto por Rahman e Dueck (2012), pode ser encontrada em um circuito reversível ternário com uma porta $\boldsymbol{G}\langle 0_x\rangle$ seguida por uma $\boldsymbol{G}\langle 1_x\rangle$, ou vice-versa, isto é, **G** é uma porta cuja variável de controle não é 2 e não se encontram na linha *x*. Este sub-circuito tem um custo igual a 8. Ele pode ser substituído por um simples gate **G** disposta na linha *y* seguido por $\boldsymbol{G^{-1}}\langle 2_x\rangle$, resultando em um custo total de 3. Por meio da Figura 36 – (b), esta redução de custo é comprovada.

Outro modelo que pode ser utilizado para a redução de custos, consiste em um determinado circuito com três portas reversíveis ternárias **G,** controladas por 0, por 1 e por 2, respectivamente. Estas portas são equivalentes a uma porta simples **G** na linha de destino (linha sem controle), como pode-se constatar na Figura 36 – (a).

Figura 36 – Modelo para a redução de custos dos circuitos.



Fonte: Próprio Autor

Uma melhoria semelhante, inferior em relação ao custo reduzido, pode ser obtida no caso de um circuito com uma porta $\boldsymbol{G}\langle 0_x\rangle$ seguida de uma $\boldsymbol{G}\langle 2_x\rangle$, com um custo igual 6. Tais portas podem ser substituídas por uma **G** simples seguido de uma $\boldsymbol{G^{-1}}\langle 1_x\rangle$ com um custo igual a 5. O custo total é reduzido em 1. O mesmo se aplica no caso de $\boldsymbol{G}\langle 1_x\rangle$ seguido de $\boldsymbol{G}\langle 2_x\rangle$, com um custo igual a 6, que pode ser substituído por um **G** simples seguido por $\boldsymbol{G^{-1}}\langle 0_x\rangle$ com um custo igual a 5.

8 CONCLUSÃO E TRABALHOS FUTUROS

As pesquisas relacionadas a circuitos reversíveis têm recebido crescente atenção desde os primeiros trabalhos motivacionais de Feynman (1986), Landauer (1961) e Bennett (1973), apontando a possibilidade de alcançar baixo consumo de energia nos microprocessadores atuais (VIERI et. al, 1998), e de trabalhar com algoritmos altamente eficientes (CHILDS; VAN DAM, 2010; FAN, 2008; WITTEK, 2014) quando, após um avanço tecnológico esperado, computadores quânticos sejam construídos (KLIMOV, 2003; TOSI et al., 2017).

Uma vez que todas as operações quânticas devem ser reversíveis, a principal aplicação dos circuitos reversíveis são consequentemente, tecnologias quânticas. Além disso, os circuitos são comumente mapeados para circuitos quânticos como forma de mensurar os custos de implementação. Quanto menor a complexidade dos circuitos reversíveis, menores serão os custos em futuras implementações. Em virtude destas características, a síntese de circuitos reversíveis ternários tem emergido como um importante problema de pesquisa.

O problema de sintetizar um circuito reversível ternário é o processo de construção de uma cascata de portas reversíveis ternárias que mapeiam cada vetor de saída para o seu vetor de entrada correspondente e exclusivo. Matematicamente, a síntese de circuito reversível representa uma decomposição da especificação do circuito, que por si só é uma permutação, para um número de permutações menores de portas reversíveis ternárias.

Neste trabalho, realizou-se a síntese de um *benchmark* completo, com todas 9! permutações possíveis para funções reversíveis ternárias de duas variáveis, empregando o método MMD *plus* nos modos *backward top-down* (versão padrão) como referência e as metodologias propostas:

I. MMD *plus* com a ordenação dos vetores de entrada por distância *Hamming*;
II. MMD *plus* com a ordenação dos vetores de entrada por peso *Hamming*;
III. MMD *plus* aplicado aos modos *forward* e *bottom*-up;
IV. MMD *plus* aplicado a ciclos de permutação de ordem natural;
V. MMD *plus* aplicado a transposições;
VI. MMD *plus* aplicado ciclos de tamanho 3.

O método MMD *plus,* cujos vetores de entrada são ordenados lexicograficamente, mostrou-se superior a Metodologia I em 12% do *benchmark*, com relação à quantidade de

portas nas realizações. Por outro lado, em relação ao custo das realizações, apresentou superioridade de 21% sobre a Metodologia I.

Ainda com relação ao custo total dos circuitos reversíveis ternários obtidos, o MMD *plus* com o emprego do modo *Top-down,* tanto na orientação de busca *backward* e *forward*, mostrou-se superior em aproximadamente 70% do *benchmark*. Em 21% do *benchmark*, os circuitos gerados com o emprego do modo *Bottom-up* mostraram-se superiores. Somente em 9% os resultados foram semelhantes.

Ademais, as realizações obtidas com os métodos MMD *plus* e MMD_T na versão bidirecional foram comparadas. O MMD *plus* determinou circuitos com no máximo 11 portas reversíveis ternárias do tipo MS, especificamente para 26 realizações, enquanto que o MMD_T para o mesmo *benchmark* produziu circuitos com até 12 portas, para 84 realizações.

Portanto, é possível concluir que as diferentes formas de ordenação e orientação para os vetores de entrada produzem circuitos, não necessariamente, com a mesma complexidade.

As realizações com base na decomposição de permutações em ciclos de diferentes comprimentos, usando como referência todo o conjunto 9! = 362880 funções reversíveis ternárias de 2 variáveis foram avaliadas e mostraram que os circuitos determinados por meio do MMD *plus,* em 54,39% do *benchmark* tiveram menor custo do que realizações baseadas em ciclos de ordem natural. Além disso, constatou-se que em 69,5% das funções de *benchmark*, ciclos não decompostos têm menor custo de realização do que os ciclos com 3 elementos. Na maioria dos casos, as realizações baseadas em 3 ciclos têm um custo menor do que as realizações baseadas em transposições.

Como os ciclos também foram realizados com o MMD *plus*, não é esperado que novas melhorias no MMD_T introduzam mudanças significativas nos resultados comparativos obtidos. A este respeito, pode-se salientar que, no caso binário, recentemente houve melhorias importantes do algoritmo MMD original. Os últimos e possivelmente mais relevantes são os de Soeken e Chattopadhyay (2015), Soeken, Dueck e Miller (2016) e Soeken (2016). Deve-se notar que em Soeken, Dueck e Miller (2016) portas quânticas controladas-V também estão incluídas, onde V denota uma porta de valor complexo, cuja especificação de matriz é igual à raiz quadrada da matriz NOT. Possíveis extensões deste tipo para o domínio ternário podem ser investigadas.

Além disso, propôs-se a síntese de circuitos reversíveis ternários com base no AGS e AGS-II , utilizando-se das portas do tipo MS. Como a estrutura de um problema de síntese é indefinida inicialmente, optou-se pelo emprego do AG como proposta de solução. O AG permite encontrar uma combinação apropriada de portas que realizam uma dada função

reversível ternária. Neste contexto, uma ampla análise sobre os impactos dos parâmetros do AG na resolução deste tipo de problema foi realizada. Por meio dos experimentos, identificou-se diferentes parâmetros para a configuração AGS e do AGS-II para os quais são produzidas melhores soluções.

8.1 TRABALHOS FUTUROS

A síntese de circuitos reversíveis é considerada uma área de pesquisa promissora, uma vez que é esperado, em um futuro próximo, a construção de computadores quânticos. Portanto, existem muitos escopos para realizar pesquisas neste contexto. Além disso, esta tese tem algumas limitações que podem também ser investigadas com maior intensidade. Uma vez que os AGs propostos leva muito tempo para encontrar soluções, diferentes estratégias poderiam ser empregadas para contornar este problema, bem como diferentes portas reversíveis ternárias poderiam ser utilizadas na determinação dos circuitos. Ademais, diferentes configurações podem ser abordadas nos AGs para explorar a possibilidade de encontrar soluções ainda melhores.

Além disso, outras propostas de trabalhos futuros podem ser consideradas, como por exemplo:

1. A implementação de um processamento pós-síntese, com base nos modelos propostos no Capítulo 7, para a otimização dos custos dos circuitos reversíveis ternários;

2. Diferentes tipos de ordenação para os vetores de entrada do MMD podem ser implementados com a finalidade de encontrar circuitos com complexidade diferente;

3. O reordenamento dos elementos dos ciclos de permutação para a obtenção de circuitos reversíveis ótimos ou próximos da otimalidade;

4. Uma extensão das metodologias de síntese propostas para circuitos reversíveis ternárias $n \times n$, isto é, com n variáveis de entradas e saídas.

REFERÊNCIAS

AHARONOV, D.; BEN-OR, M. Fault-tolerant quantum computation with constant error rate. **SIAM Journal on Computing**, Nova Zelândia, v. 38, n. 4, p. 1207-1282, 2008.

BABU, H. M.H. et al. Synthesis of full-adder circuit using reversible logic. In: INTERNATIONAL CONFERENCE ON VLSI DESIGN, 17., 2004, Mumbai, India. **Proceedings....** [S.l.]: IEEE, 2004. p. 757-760.

BARENCO, A. et al. Elementary gates for quantum computation. **Physical Review**, [S.1.], v. 52, n. 5, p. 3457, 1995.

BASU, S. et al. An efficient synthesis method for ternary reversible logic. In: CIRCUITS AND SYSTEMS (ISCAS), 2016, Montreal, QC, Canada. **Proceedings…** [S.l.]: IEEE, 2016. p. 2306-2309.

BAUMSLAG, B.; CHANDLER, B. **Theory and problems of group theory.** New York: Mcgraw-hill, 1968. 278 p.

BENNETT, C. Logical Reversibility of Computation. **IBM Journal of Research and Development**, Nova York, v. 17, n. 6, p. 525 - 532, Novembro 1973.

BERNSTEIN, D. S. **Matrix mathematics: theory, facts, and formulas with application to linear systems theory**. Princeton: Princeton University Press, 2005.

CHILDS, A. M.; VAN DAM, W. Quantum algorithms for algebraic problems. **Reviews of Modern Physics**, Minneapolis, v. 82, n. 1, p. 1, 2010.

DATTA, K. et al. An evolutionary approach to reversible logic synthesis using output permutation. In: INTERNATIONAL DESIGN AND TEST SYMPOSIUM (IDT), 8 th. 2013, Marrakesh. **Proceedings…** [S.l.]: IEEE, 2013. p. 1-6.

DE VOS, A.; RAA, B.; STORME, L. Generating the group of reversible logic gates. **J. of Physics A: Mathematical and General**, Bristol, v. 35, p. 7063–7078, 2002.

DE VOS, A. Reversible computer hardware. **Electronic Notes in Theoretical Computer Science**, Iorque, Inglaterra, v. 253, n. 6, p. 17-22, 2010.

DE VOS, A.; BURIGNAT, S.; THOMSEN, M. Reversible implementation of a discrete linear transformation. In: 2ND WORKSHOP ON REVERSIBLE COMPUTATION (RC 2010). Bremen. **Proceedings....** Bremen: Lecture Notes in Computer Science (LNCS), 2010. p. 107-110.

DEIBUK, V.; BILOSHYTSKYI, A. Genetic synthesis of new reversible/quantum ternary comparator. **Advances in Electrical and Computer Engineering**, România, v. 15, n. 3, p. 147-152, 2015.

DRECHSLER, R.; FINDER, A.; WILLE, R. Improving ESOP-based synthesis of reversible logic using evolutionary algorithms. In: EUROPEAN CONFERENCE ON THE APPLICATIONS OF EVOLUTIONARY COMPUTATION, 2011, Berlin. **Proceedings…** Berlin: Springer, 2011. p. 151-161.

FAN, Y. A Generalization of the deutsch-jozsa algorithm to multi-valued quantum logic. In: INTERNATIONAL SYMPOSIUM ON MULTIPLE-VALUED LOGIC, 37., Oslo, Norway. **Proceedings...** Oslo, Norway: IEEE, 2007. p. 1-12.

FEYNMAN, R. P. Quantum mechanical computers. **Foundations of Physics**, New York, v. 16, n. 6, p. 507-531, 1986.

FONSECA, C. M. M. da. **Multiobjective genetic algorithms with application to control engineering problems**. 1995. 167 f. Tese (Doutorado), University of Sheffield, Sheffield, 1995.

GOLDBERG, D. E. **Genetic algorithms in search, optimization, and machine learning.** Boston, USA: Addison Wesley, 1989.

GOLUBITSKY, O.; FALCONER, S. M.; MASLOV, D. Synthesis of the optimal 4-bit reversible circuits. IN: AUTOMATION CONFERENCE (DAC), 47TH., 2010, Anaheim. **Proceedings...** Anaheim: IEEE, 2010. p. 653-656.

HADJAM, F. Z.; MORAGA, C. RIMEP2: Evolutionary design of reversible digital circuits. **ACM Journal on Emerging Technologies in Computing Systems (JETC)**, Nova York, v. 11, n. 3, p. 27, 2014.

HU, Z.; DEIBUK, V. New design of reversible/quantum devices for ternary arithmetic. In: DATA STREAM MINING & PROCESSING (DSMP), 2016, Ucrânia. **Proceedings...** Ucrânia: IEEE, 2016. p. 80-84.

KHAN, M. H. Design of reversible/quantum ternary comparator circuits. **Engineering Letters**, São Francisco, v. 16, n. 2, p. 1-7, 2008.

KHAN, M. H.; PERKOWSKI, M. A. Quantum ternary parallel adder/subtractor with partially-look-ahead carry. **Journal of Systems Architecture**, Nova York, v. 53, n. 7, p. 453-464, 2007.

KHAN, M. H.; PERKOWSKI, M. Genetic algorithm based synthesis of multi-output ternary functions using quantum cascade of generalized ternary gates. In: EVOLUTIONARY COMPUTATION, 2004. CEC2004**,** Portland**. Proceedings...** Portland: IEEE, 2004. p. 2194-2201.

KHANOMA, R.; KAMALB, T.; KHANA, M. H. Genetic algorithm based synthesis of ternary reversible/quantum circuit. In: COMPUTER AND INFORMATION TECHNOLOGY, 2008, Khulna, Bangladesh. **Proceedings...** Khulna, Bangladesh: IEEE, 2008. p. 270-275.

KLIMOV, A. B. et al. Qutrit quantum computer with trapped ions. **Physical Review A**, Austin, Texas, v. 67, n. 6, p. 062313, 2003.

KLIMOV, A. B. et al. Quantum phases of a qutrit. **Journal of Physics A: Mathematical and General**, [S.1] v. 37, n. 13, p. 4097, 2004.

KOLE, A. et al. Exact synthesis of ternary reversible functions using ternary toffoli gates. In: MULTIPLE-VALUED LOGIC (ISMVL), 2017., Novi Sad, Serbia. **International Symposium...** Novi Sad, Serbia: IEEE, 2017. p. 179-184.

LI, X.; YANG, G.; ZHENG, D. Logic Synthesis of Ternary Quantum Circuits with Minimal Qutrits. **JCP**, Los Angeles, v. 8, n. 8, p. 1941-1946, 2013.

LUKAC, M.; PERKOWSKI, M. Combining evolutionary and exhaustive search to find the least expensive quantum circuits. In: PROCEEDINGS OF ULSI SYMPOSIUM, 2005, Honolulu, Hawaii. **Proceedings...** Honolulu, Hawaii: 2005. p. 33-45.

LUKAC, M. et al. Evolutionary approach to quantum and reversible circuits synthesis. **Artificial Intelligence Review**, Norwell, v. 20, n. 3-4, p. 361-417, 2003.

LUKAC, M.; PERKOWSKI, M.; KAMEYAMA, M. Evolutionary quantum logic synthesis of boolean reversible logic circuits embedded in ternary quantum space using structural restrictions. In: EVOLUTIONARY COMPUTATION (CEC), 2010, Barcelona. **Proceedings...** Barcelona: IEEE, 2010. p. 1-8.

Khan, M. H. A. GFSOP-based ternary quantum logic synthesis. In: OPTICS AND PHOTONICS FOR INFORMATION PROCESSING IV, Naha, Okinawa. **Proceedings...** Japão: IEEE, 2009. p. 103-108.

MANDAL, S. B.; CHAKRABARTI, A.; SUR-KOLAY, S. Synthesis techniques for ternary quantum logic. In: MULTIPLE-VALUED LOGIC (ISMVL), 41st., 2011, Pistacaway. **Porceedings…** Pistacaway: IEEE, 2011. p. 218-223.

MILLER, D.**;** MASLOV, M. D.; DUECK, G. W. Synthesis of quantum multiple-valued circuits. **Journal of Multiple Valued Logic and Soft Computing,** Philadelphia, v. 12, n. 50-6, p. 431, 2006.

MOHAMMADI, M. Efficient genetic based methods for optimizing the reversible and quantum logic circuits. **Journal of Advances in Computer Research**, Iran, v. 3, n. 3, p. 85-96, 2012.

MONFARED, A. T.i; HAGHPARAST, M. Design of new quantum/reversible ternary subtractor circuits. **Journal of Circuits, Systems and Computers**, Singapore, v. 25, n. 02, p. 1650014, 2016.

MOORE, G. E. Cramming more components onto integrated circuits. **IEEE Solid-State Circuits Society Newsletter**, São Francisco, v. 11, n. 3, p. 33-35, 2006.

MORAGA, C. Design of p-valued deutsch quantum gates with multiple control signals and mixed polarity. In: INTERNATIONAL CONFERENCE ON REVERSIBLE COMPUTATION, 2016, Bologne, Itália. **Proceedings...** Bologne: Springer, 2016. p. 175-180.

MORAGA, C. Quantum p-valued toffoli and deutsch gates with conjunctive or disjunctive mixed polarity control. In: MULTIPLE-VALUED LOGIC (ISMVL), 2016, Sérbia. **Proceedings...** Sérbia: IEEE, 2016. p. 241-246.

N, M.; RICE, J. E. Synthesis of reversible logic functions using ternary max-min algebra. In: CIRCUITS AND SYSTEMS (ISCAS), 2016, Montreal. **Proceedings...** Montreal: IEEE, 2016. p. 1674-1677.

ALHAGI, N.; HAWAS, M.; PERKOWSKI, M. Synthesis of Reversible Circuits with no Ancilla Bits for Large Reversible Functions. In: INTERNATIONAL SYMPOSIUM ON MULTIPLE-VALUED LOGIC, BARCELONA, 2010, Espanha. **Proceedings...** Espanha: IEEE, 2010. p. 26-28.

NAGAMANI, A. N. et al.; A Genetic algorithm based heuristic method for test set generation in reversible circuits. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, California, v. 37, n. 2. p. 324-336, 2018.

NIELSEN, M. A.; CHUANG, Isaac. **Quantum computation and quantum information.** Nova York, EUA: Cambridge University Press, 2002. p. 1-676.

PAULI, W.; Die allgemeinen prinzipien der wellenmechanik. **Handbuch der Physik**, Berlim, v. 5, p. 1-168, 1958.

RAHMAN, M.; DUECK, G. W. An algorithm to find quantum templates. In: EVOLUTIONARY COMPUTATION (CEC), 2012, Brisbane. **Proceedings...** Brisbane: IEEE, 2012. p. 1-7.

RANI, P. M. et al. Improved decomposition of multiple-control ternary toffoli gates using muthukrishnan-stroud quantum gates. In: INTERNATIONAL CONFERENCE ON REVERSIBLE COMPUTATION, 2017, Cham. **Proceedings...** Cham: Springer, 2017. p. 202-213.

SASANIAN, Zahra et al. A cycle-based synthesis algorithm for reversible logic. In: SOUTH PACIFIC DESIGN AUTOMATION CONFERENCE, 09., Yokohama, Japan. **Proceedings...** New York: IEEE Press, 2009. p. 745-750.

SASANIAN, Z. **Technology mapping and optimization for reversible and quantum circuits**. 2012. 145 f. Tese (Doutorado em Filosofia) – Faculdade de Ciência da Computação, Universidade de Victoria, Victoria, 2012.

SEDGEWICK, R. Permutation generation methods. **ACM Computing Surveys (CSUR)**, Nova York, v. 9, n. 2, p. 137-164, 1977.

SHOR, P. W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. **SIAM review**, Philadelphia, v. 41, n. 2, p. 303-332, 1999.

SOEKEN, M. et al. An extension of transformation-based reversible and quantum circuit synthesis. In: CIRCUITS AND SYSTEMS (ISCAS), 2016, Montreal. **Proceedings...** Montreal: IEEE, 2016. p. 2290-2293.

SOEKEN, M.; CHATTOPADHYAY, A. Fredkin-enabled transformation-based reversible logic synthesis. In: MULTIPLE-VALUED LOGIC (ISMVL), 2015. Waterloo. **Proceedings...** Waterloo: IEEE, 2015. p. 60-65.

SOEKEN, M.; DUECK, G. W.; MILLER, D. A fast symbolic transformation based algorithm for reversible logic synthesis. In: INTERNATIONAL CONFERENCE ON REVERSIBLE COMPUTATION, 2016, Bologna. **Proceedings...** Bologna: Springer International Publishing, 2016. p. 307-321.

TOSI, G. et al. Silicon quantum processor with robust long-distance qubit couplings. **Nature Communications...** Reino Unido: Nature, 2017. Disponível em: <https://phys.org/news/2017-09-flip-flop-qubits-radical-quantum.html>. Acesso em: 02 jun. 2018.

VIERI, C. et al. A fully reversible asymptotically zero energy microprocessor. In: POWER DRIVEN MICROARCHITECTURE WORKSHOP, 1998, Barcelona. **Proceedings...** Barcelona, 1998, p. 138-142.

WANG, X. et al. A variable-length chromosome evolutionary algorithm for reversible circuit synthesis. **Journal of Multiple-Valued Logic & Soft Computing,** v. 25, n. 6, p. 643-671, 2015.

WEI, V. K. Generalized hamming weights for linear codes. **IEEE Transactions on information theory**, Nova Jersey, v. 37, n. 5, p. 1412-1418, 1991.

WITTEK, P. **Quantum machine learning: what quantum computing means to data mining.** San Diego: Elsevier, 2014. 176 p.

YANG, G. et al. Realizing ternary quantum switching networks without ancilla bits. **Journal of Physics A: Mathematical and General**, Reino Unido, v. 38, n. 44, p. 9689-9697, 2005.

ZADEH, R.; HAGHPARAST, M. A new reversible/quantum ternary comparator. **Australian Journal of Basic and Applied Sciences**, Australia, v. 5, n. 12, p. 2348-2355, 2011.

ZOBOV, V.; PEKHTEREV, D. I. Adder on ternary base elements for a quantum computer. **JETP letters**, Rússia, v. 89, n. 5, p. 260-263, 2009.