

Universidade Federal de Ouro Preto - UFOP
Instituto de Ciências Exatas e Biológicas - ICEB
Departamento de Ciência da Computação - DECOM

**Algoritmos Heurísticos e Metaheurísticos para a
Minimização do Custo de Conexões em Matrizes de Portas**

Bolsista: João Vitor Mascarenhas dos Santos

Orientador: Marco Antonio M. Carvalho – DECOM/UFOP

Nota: Relatório Final referente ao período de 01/04/2014 a 29/08/2014, apresentado à Universidade Federal de Ouro Preto, como parte das exigências do programa de iniciação científica -FAPEMIG.

Local: Ouro Preto - Minas Gerais - Brasil

Data: 29 de agosto de 2014

Título do Resumo

Assinatura do orientador(a): -----
Nome Completo do orientador(a)

Assinatura do co-orientador(a): -----
Nome Completo do co-orientador(a)

Assinatura do bolsista: -----
Nome Completo do bolsista

Sumário

1	Introdução	1
1.1	Descrição do Problema	2
2	Objetivos	5
2.1	Objetivos Específicos	5
3	Fundamentos Teóricos e Revisão	6
4	Materiais e Métodos	9
5	Resultados e Discussões	10
6	Conclusão	11

Capítulo 1

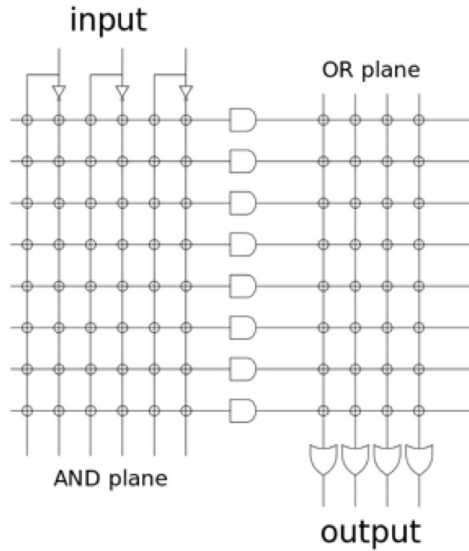
Introdução

A microeletrônica é um ramo da eletrônica, voltado à integração de circuitos eletrônicos, promovendo uma miniaturização dos componentes em escala microscópica. A área engloba tanto os processos físico-químicos de fabricação dos circuitos integrados como o projeto do circuito em si. São considerados ramos desta área, igualmente, o desenvolvimento de *software* de apoio ao projeto de circuitos, modelagem de componentes, técnicas de teste, entre outras.

O componentes utilizados na microeletrônica são construídos na escala de microns ou mesmo nanômetros, tornando-se parte do ramo de nanotecnologia. O conjunto de componentes usados para um mesmo projeto é tipicamente chamado de circuito integrado (CI), ou ainda, *chip*. Alguns exemplos de circuitos integrados são memórias de computadores, processadores, modems e conversores analógicos digitais.

Os circuitos integrados são produzidos em *wafers*, discos de silício, normalmente de 300mm de diâmetro, sobre os quais são fabricadas estruturas laminares, em um processo denominado fotolitografia. Tipicamente, diversos circuitos eletrônicos (idênticos ou não) são fabricados em um mesmo *wafer* por vez, de modo a ocupar toda a área disponível. O custo de produção do circuito será o custo de fabricação do *wafer*, dividido pelo número de circuitos nele presentes (excluindo-se os defeituosos). Assim, um circuito é mais caro pelo fato de usar uma maior área do *wafer* – e, conseqüentemente, tendo um menor número de circuitos por *wafer* – e não necessariamente devido ao número de componentes presentes. Circuitos de leiaute otimizado são menores, e por consequência, mais baratos, uma vez que ocupam uma menor área do *wafer*.

Uma *Matriz Lógica Programável* (MLP) ou *Matriz de Portas* é um tipo de dispositivo lógico programável usado para implementar circuitos lógicos combinacionais. Uma MLP possui uma dimensão com portas AND programáveis que podem ser conectadas a uma dimensão de portas OR igualmente programáveis. Desta forma, conexões entre portas podem ser estabelecidas para produzir um resultado específico. A Figura 1 apresenta a disposição das dimensões (ou planos) de uma MLP, o que permite que um grande número de funções lógicas sejam sintetizadas.



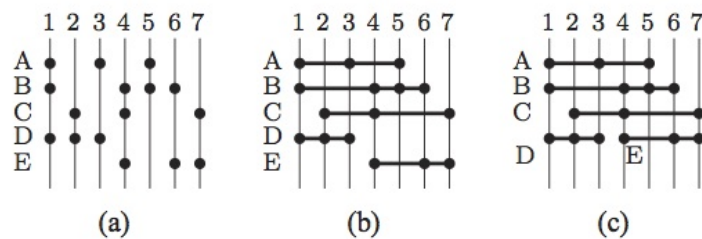
Exemplo de Matriz Lógica Programável, ou Matriz de Portas.

O *Problema de Minimização de Custos de Conexões em Matrizes de Portas* (PMCCMP) é definido como o problema de determinar uma permutação de portas de maneira tal que o número necessário de trilhas é minimizado e consequentemente, a área e o custo de produção do circuito correspondente sejam minimizados.

1.1 Descrição do Problema

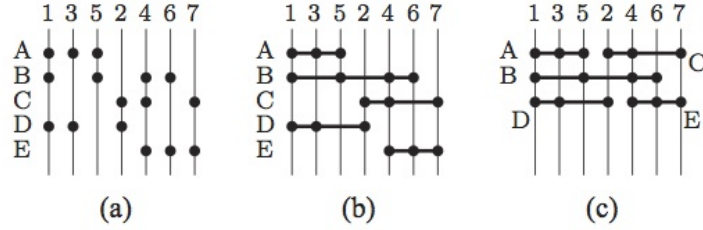
Em um circuito eletrônico impresso implementado por uma matriz lógica programável, as portas correspondem a *nós*, e diferentes conexões são exigidas entre eles. Cada conexão é realizada por um *fio* envolve um subconjunto de nós e é chamada de rede. A Figura 2(a) apresenta um exemplo de matriz porta em que sete portas (linhas verticais) precisam ser conectadas de acordo com cinco redes diferentes, representadas por linhas em uma mesma linha: a rede *A* conecta as portas 1, 3 e 5; a rede *B* conecta as portas 1, 4, 5 e 6, e assim por diante. Fios condutores são utilizados para criar as conexões, um para cada rede, conforme mostrado nas linhas horizontais na Figura 2(b).

É importante notar que para conectar as portas de uma rede, pode ser necessário cruzar outras portas não incluídas na rede, dependendo da disposição das portas. Também, uma única *trilha* de conexão pode ser utilizada para posicionar fios sem sobreposição, conforme exibido na Figura 2(c) para as redes *D* e *E*. O comprimento total dos fios utilizados determina o custo das conexões, ao passo que o número de trilhas determina a área total do circuito eletrônico, o que pode ser limitado de acordo com restrições de projeto. Ambos indicadores fornecem uma estimativa da eficiência do leiaute do circuito e dependem de como as portas são sequenciadas fisicamente.



Exemplo de Matriz de Portas: (a) conexões exigidas, (b) redes já com fios e (c) trilhas de conexão.

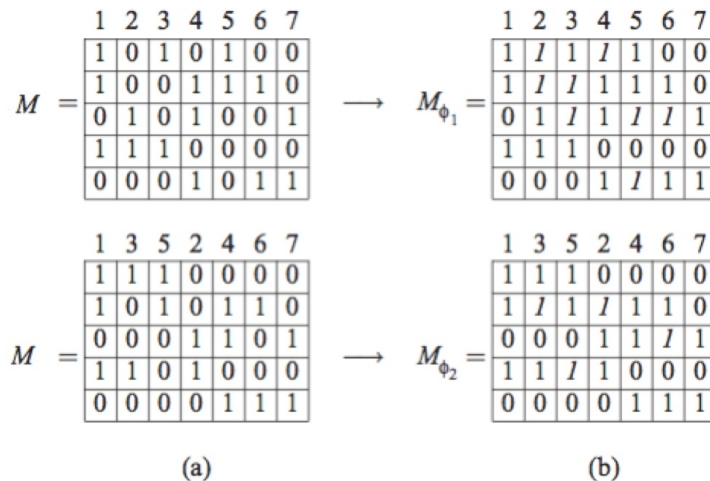
O leiaute exibido na Figura 2 exige 19 unidades de fios e 4 trilhas, correspondente ao número máximo de trilhas utilizadas. Um leiaute otimizado é apresentado na Figura 3, usando 15 unidades de fios e 3 trilhas. Note que a diferença entre o leiaute da mesma matriz de portas original é a disposição das portas. Na Figura 2 temos a sequência 1, 2, 3, 4, 5, 6 e 7, ao passo que na Figura 3 temos a sequência 1, 3, 5, 2, 4, 6 e 7.



Leiaute otimizado da Matriz de Portas anterior.

Uma instância do PMCCMP pode ser descrita por uma matriz $M \in \{0, 1\}^{m \times n}$ e um parâmetro $\lambda \in \mathbb{Z}_+$ representando o número de trilhas disponíveis e, portanto, um limitante superior para o número máximo de trilhas necessárias, indicando que quaisquer permutações com número de trilhas maior que λ são inviáveis. As linhas da matriz M são associadas às m redes e as colunas são associadas às n portas e $M(i, j) = 1$ se e somente se a rede i incluir a porta j . Uma solução para o PMCCMP em uma sequência $\phi : [1, \dots, n] \rightarrow [1, \dots, n]$ em que $\phi(j)$ indica posição da porta j na matriz de portas. Tal solução define uma nova matriz M_ϕ , obtida pela permutação das colunas de M de acordo com ϕ . A matriz M_ϕ é possuidora da propriedade dos 1s consecutivos, desta forma que $M_\phi(i, j) = 1$ se e somente se, de acordo com ϕ , o fio da rede i incluir ou cruzar a porta j .

A Figura 4 apresenta a matriz M correspondente à matriz de portas da Figura 2(a) e as matrizes M_{ϕ_1} e M_{ϕ_2} das sequências $[1, 2, 3, 4, 5, 6, 7]$ e $[1, 3, 5, 2, 4, 6, 7]$. Note que os valores para o máximo de trilhas e o comprimento total dos fios podem ser facilmente obtidas a partir de M_ϕ . O comprimento do fio da rede i é a distância (em número de portas) entre a primeira e última portas de i , igual ao número de 1's na i -ésima linha de M_ϕ menos 1 (não se considera a primeira porta), logo, o comprimento total dos fios de uma matriz de portas é a soma de todos os elementos não nulos de M_ϕ menos m . O número máximo de trilhas é o número máximo de 1's em qualquer uma das colunas de M_ϕ .



Matrizes M (a) e matrizes M_{ϕ_1} e M_{ϕ_2} das sequências [1, 2, 3, 4, 5, 6, 7] e [1, 3, 5, 2, 4, 6, 7], com os elementos trocados em *itálico*.

Em suma, dada uma matriz $M \in \{0, 1\}^{m \times n}$ e um limite $\lambda \in \mathbb{Z}_+$ o objetivo do PMCCMP é determinar uma permutação das colunas que resulte em uma matriz M_ϕ correspondente com menor soma de elementos em qualquer coluna, sendo este valor menor do que λ .

Este é um problema NP-Difícil (Linhares e Yanasse, 2002) e possui aplicações em diferentes áreas, bem como diferentes problemas de formulação equivalente (Mohring, 1990): Problema de Corte Modificado (*Modified Cutwidth*), Problema de Minimização de Pilhas Abertas (*Minimization of Open Stacks Problem*), Dobradura de Arranjos Lógicos Programáveis (*Programmable Logic Array Folding*, ou PLA Folding), *Interval Thickness*, *Node Search Game*, *Edge Search Game*, *Narrowness*, *Split Bandwidth*, *Graph Pathwidth*, *Edge Separation* e *Vertex Separation*.

Capítulo 2

Objetivos

1. Elaborar heurísticas consistentes e robustas que possam ser utilizadas no contexto de problemas de leiaute de matrizes de portas que permitam a obtenção rápida de soluções próximas da solução ótima sem que se perca a vantagem da busca sistemática – inicialmente considerando problemas específicos, mas com uma possibilidade de generalização;
2. Avaliar o comportamento heurístico de diferentes formulações para problemas de sequenciamento;
3. Pesquisar técnicas para melhoria fina de soluções obtidas por heurísticas usando programação linear inteira;
4. Ampliar o conjunto de problemas testes com soluções ótimas já conhecidas através da execução de testes sistemáticos usando formulações já desenvolvidas;
5. Será buscada ainda a aplicação prática dos métodos desenvolvidos em contextos reais, a fim de que também seja constituído um avanço para as indústrias nacionais;
6. Além dos objetivos principais, outros produtos deste projeto de pesquisa serão trabalhos publicados em periódicos e eventos nacionais e internacionais, os quais contribuem para a promoção dos centros de pesquisas nacionais e também da tecnologia.

2.1 Objetivos Específicos

1. Propor um modelo de otimização que contemple apropriadamente as especificidades de diferentes aplicações do problema com o uso dos modelos descritos na literatura;
2. Implementação computacional de um software para determinação do leiaute de matrizes de portas usando ferramentas de Inteligência Computacional, o que inclui a utilização de métodos heurísticos e metaheurísticos;
3. Avaliação do software implementado considerando dados reais e também com problemas teste publicamente disponíveis;

Capítulo 3

Fundamentos Teóricos e Revisão

Esta seção apresenta os trabalhos mais relevantes da literatura dos problemas relacionados ao leiaute de matrizes de portas.

O problema de leiaute de matriz de portas (GMLP) foi provado ser NP-difícil (Kashiwabara e Fujisawa, 1979) por uma redução do problema de aumento de um grafo de intervalos (IGAP – *interval graph augmentation problem*).

Considerando os autores que aplicaram métodos heurísticos para a resolução do GMLP, cita-se o trabalho seminal de Wing, Huang e Wang (1985) que também trataram o net assignment. Neste trabalho o GMLP foi modelado usando grafos de intervalos, considerando portas em comum em diferentes trilhas como uma interseção. Estas trilhas com portas em comum podem ser representadas por uma única trilha, diminuindo a área necessária para implementação do circuito. Como experimento, obtiveram leiautes para circuitos CMOS de 180 e 306 transistores em 9 e 28 segundos, respectivamente, em um computador mainframe. No entanto, Hwang, Fuchs e Kang (1987) conseguiram resultados mais relevantes com o algoritmo *min-net-cut* modificado em conjunto com *dynamic-net-lists* e complexidade de tempo $O(n \log n)$, em que n representa o número total de transistores e conexões de portas. Tais resultados apresentados foram a diminuição do número de trilhas no circuito em relação ao trabalho anterior, gerando um leiaute com 11 trilhas a menos na instância de 360 transistores e 71 portas. O algoritmo *min-net-cut* foi utilizado novamente por Chen e Hou (1988), que juntamente com uma representação dinâmica das ligações de um circuito criaram um algoritmo para o GMLP que analisa a dualidade do CMOS. O algoritmo, cuja complexidade de tempo é $O(n \log n)$, em que n é o número de portas, apresentou melhores resultados que os anteriores da literatura nos 5 circuitos testados. Já Singh e Chen (1990) modelaram o problema utilizando multigrafos e equações lógicas para determinar a sequência das portas. Os experimentos computacionais reportados consideraram 5 instâncias e incluíram o algoritmo de Hwang, Fuchs e Kang (1987), o qual foi superado em todas as instâncias. Chen e Hu (1990a, 1990b) apresentaram dois algoritmos, *GM_Plan* e *GM_Learn*, ambos baseados em paradigmas de inteligência artificial que alcançaram, ou determinaram, os melhores resultados da literatura até aquele momento. Estes mesmos trabalhos abordaram as mesmas instâncias utilizadas por Deo, Krishnamoorthy e Langston (1987), Huang e Wing (1989), Heinbuch (1988), Chang, Chang e Hsu (1987), Wing (1982), Wing e Huang (1985), Li (1983), Leong (1986), Nakatani et al. (1986), Wing (1983). No final da década de 80 e boa parte da década de 90 o GMLP foi abordado predominantemente por técnicas metaheurísticas, relacionadas na sequência. Após este período, Singh e Chen (1992) voltam a utilizar o algoritmo *mincut* na implementação de uma solução para o GMLP. Comparações quanto ao número de trilhas foram feitas com a implementação do algoritmo apresentado por Hwang, Fuchs e Kang (1987), que também é utiliza o *mincut*. Apesar de

conseguirem melhores resultados em todas as instâncias, o algoritmo apresentado neste artigo demora mais tempo para alcançar a solução.

Shahookar et al. (1993) propôs uma nova implementação de algoritmo genético em conjunto com *beam search* a qual foi denominada *Genetic Beam Search*. O algoritmo genético do pacote *Genesis* (Grefenstette e Schraudolph, 1987) foi utilizado e adaptado por funções de avaliação apropriadas para implementar o *beam search*. Considerando três instâncias que também foram utilizadas por Hong, Park e Kim (1989), foi possível diminuir o número de trilhas em um caso e obter menor comprimento de rede em todos.

Com a estratégia de busca predatória, a qual restringe a área de busca e a cada nova melhoria encontra uma solução, Linhares (1999) alcançou os resultados obtidos por Chen e Hu (1990a, 1990b). Além disso, das 25 instâncias, em 12 conseguiu igualar o número de trilhas ao limite inferior, obtido pela seleção da maior soma de coluna da matriz original, e em 4 conseguiu os melhores resultados da literatura, para os outros casos ele conseguiu alcançar os resultados obtidos pelo *GM Plan* e o *GM Learn*. Linhares, Yanasse e Torreão (1999) apresentaram a otimização microcanônica, um procedimento derivado da física estatística assim como o *simulated annealing*, no entanto, considera um sistema termicamente isolado. Dentre os 30 circuitos testados, o método foi capaz de igualar todos os resultados anteriores de Hong, Park e Kim (1989) nos circuitos lógicos unidimensionais e de Chen e Hu (1990a, 1990b) nos bidimensionais, dos quais em 5 superou com grande vantagem os melhores leiautes.

O algoritmo genético foi utilizado novamente por Oliveira e Lorena (2002). Foi proposta uma abordagem construtiva que, considerando as onze instâncias introduzidas por Chen e Hu (1990a, 1990b) e Linhares, Yanasse e Torreão (1999), alcançou todos os melhores resultados conhecidos da literatura, obtidos anteriormente pela otimização microcanônica de Linhares, Yanasse e Torreão (1999). Segundo os autores este algoritmo genético construtivo é mais robusto do que a otimização microcanônica.

Uma abordagem evolutiva de múltiplas populações foi introduzida por Mendes et al. (2002), que utilizou um algoritmo memético como motor de busca. Este trabalho igualou os resultados da busca predatória de Linhares (1999). No entanto, o critério de parada do algoritmo foi o tempo de execução que variou de 30 segundos a 60 minutos. Já a abordagem de população múltipla de Mendes e Linhares (2004), agregou algoritmos genético e memético. Este trabalho apresentou resultados iguais aos da otimização microcanônica de Linhares, Yanasse e Torreão (1999), superando todas as abordagens metaheurísticas e heurísticas anteriores da literatura – incluindo *simulated annealing*, *microcanonical annealing*, *GM-Plan*, *GM-Learn*, e heurísticas construtivas – tanto em qualidade das soluções obtidas quanto em eficiência, em termos de convergência do algoritmo. A rápida convergência é justificada pela adição de uma busca local heurística que proporcionou a redução do espaço de busca, considerando apenas o gargalo do problema, o que foi denominado colunas críticas. A única diferença de desempenho entre este algoritmo e a otimização microcanônica de Linhares, Yanasse e Torreão (1999) está na alta eficiência, em termos do número de avaliações, sendo que tal redução de esforço varia entre 80% e 90%, dependendo do tamanho da instância.

Por fim, Giovanni et al. (2013) considerou não só o GMLP, mas também o problema de minimização de custo de conexões do GMLP, e problemas correlatos, como o tempo e o número máximo de pilhas abertas. Dois algoritmos foram utilizados, sendo o primeiro uma heurística baseada em uma abordagem genética em conjunto com a definição composta e dinâmica da função de *fitness* (GAG), e o segundo, um *branch-and-cut*, que explora a flexibilidade de uma nova formulação de programação inteira com base na propriedade dos 1's consecutivos em matrizes. Foram utilizadas as 330 instâncias de Chen e Hu (1990a), SCOOP Team (2009), Faggioli e Bentivoglio (1998) e Smith e Gent (2005). Em comparação com Lorena (2002), nas

11 instâncias VLSI de Chen e Hu (1990a), o método proposto apresentou uma performance melhor. Especificamente nas instâncias do MOSP foram obtidos resultados ruins com gap alto ou sequer resolveram a relaxação linear em uma hora.

Os métodos exatos aplicados ao GMLP são encontrados na literatura com frequência muito menor. Destaca-se a formulação por programação dinâmica apresentada em Deo, Krishnamoorthy e Langston (1987). No entanto, não foram realizados experimentos computacionais que permitissem aferir a qualidade da formulação. Neste mesmo trabalho o autor prova não haver algoritmo de aproximação absoluta para o GMLP.

Capítulo 4

Materiais e Métodos

Nesta primeira etapa do projeto, foi realizada uma revisão sistemática da literatura a respeito do problema tratado e também dos problemas relacionados, conforme descrito no capítulo anterior. Todos os trabalhos foram classificados de acordo com o problema tratado, abordagem e data. Desta forma, foi traçado um panorama acurado do estado da arte da pesquisa relacionada ao problema tratado.

Ainda, foram coletadas instâncias e geradores de instâncias adotados pela comunidade acadêmica para o problema tratado, recursos estes que serão utilizados em fases subsequentes do projeto.

Ainda faltando três quartos da duração do projeto, o material gerado durante o período aqui relatado será utilizado para desenvolver uma ferramenta de definição de leiaute de circuitos eletrônicos. Esse *software* irá incorporar as tecnologias estado-da-arte em otimização para a resolução de um modelo construído com base em experiências encontradas na literatura e eventualmente em estudos em empresas que estabeleçam um convênio com a universidade. Os métodos desenvolvidos permitirão que a produção de circuitos eletrônicos seja mais eficiente, gerando melhor aproveitamento dos recursos disponíveis e menor custo associado.

Após a validação dos resultados obtidos, será realizada uma avaliação crítica a respeito do trabalho desenvolvido, abrangendo todas as estruturas e estratégias de solução empregadas, a fim de analisar o desempenho individual de cada uma.

Os resultados obtidos pelos métodos propostos serão comparados com resultados obtidos por métodos que compõem o estado da arte em relação a cada problema específico. Na comparação, serão analisados e discutidos a relação entre o desempenho dos métodos propostos e as propriedades de cada solução obtida.

Os algoritmos e programas desenvolvidos serão documentados de forma apropriada e as propostas e resultados experimentais serão divulgados para a comunidade científica.

Capítulo 5

Resultados e Discussões

O cronograma do projeto foi cumprido rigorosamente. Conforme apresentado no Capítulo 4, uma revisão sistemática da literatura foi realizada, sendo gerada toda a base conceitual para o novo aluno que substituirá o bolsista.

Um relatório técnico foi gerado e será aproveitado em futuras publicações relacionadas ao tema ao longo da execução do projeto.

Os produtos futuros deste projeto de pesquisa serão também avaliados por pares (comitês científicos e corpos editoriais) na forma de artigos submetidos para periódicos nacionais e internacionais. Um artigo científico será submetido para uma revista indexada com reconhecida relevância na área e também será submetido ao prêmio de iniciação científica promovido pela Sociedade Brasileira de Pesquisa Operacional. Como resultados adicionais esperados temos a participação em congressos e conferências onde os temas pesquisados serão discutidos com a comunidade científica.

Capítulo 6

Conclusão

Este relatório final de iniciação científica relatou o trabalho de pesquisa realizado durante quatro meses de duração de parte de um projeto. O cronograma do projeto foi cumprido rigorosamente, sendo gerada toda a base conceitual para as etapas seguintes. Um relatório técnico foi gerado e será aproveitado em futuras publicações relacionadas ao tema ao longo da execução do projeto.