

Programação matemática e método heurístico para o *floorplanning* de circuitos VLSI como um problema de empacotamento

Letícia Leite Pavanello

Universidade Estadual Paulista (UNESP)

Bauru/SP - Brasil

leticia.pavanello@unesp.br

Carlos Diego Rodrigues

Universidade Federal do Ceará (UFC)

Fortaleza/FC - Brasil

diego@lia.ufc.br

Adriana Cristina Cherri

Universidade Estadual Paulista (UNESP)

Bauru/SP - Brasil

adriana.cherri@unesp.br

RESUMO

Este trabalho aborda o *floorplanning* de circuitos integrados VLSI como um problema de empacotamento com retângulos flexíveis. Neste problema, módulos que podem ter dimensões flexíveis devem ser alocados e interconectados em um chip, respeitando determinadas condições. Além disso, a alocação deve ser feita de forma a minimizar o comprimento do fio utilizado nas conexões. Um modelo matemático e uma abordagem heurística são propostos para resolver o *floorplanning* de circuitos VLSI. Combinando programação matemática e um método heurístico, um procedimento iterativo de janela deslizante utiliza diferentes critérios de ordenação para organizar os módulos a serem alocados. Para avaliar o desempenho do método, foram realizados testes computacionais com instâncias MCNC em C++ com o solver CPLEX.

PALAVRAS CHAVE. Problema de empacotamento. Circuito VLSI. Retângulos flexíveis.

ABSTRACT

This work addresses the floorplanning in VLSI integrated circuits as a packaging problem with soft rectangles. In this problem, modules that can have flexible dimensions must be allocated and interconnected on a chip, respecting certain conditions. In addition, the allocation must be done in a way that minimizes the length of wire used in the connections. A mathematical model and a mathematical approach are proposed to solve the floorplanning of VLSI circuits. Combining mathematical programming and a heuristic method, an iterative sliding window procedure uses different sorting criteria to organize the modules to be allocated. To evaluate the performance of the method, computational tests were performed with MCNC instances in the C++ programming language with the CPLEX solver.

KEYWORDS. Packing Problem. VLSI Circuit. Soft Rectangles.

1. Introdução

O VLSI é uma tecnologia de microeletrônica que integra uma grande quantidade de módulos eletrônicos numa pastilha de silício, conhecida como chip. Com o avanço da tecnologia de circuito integrado e a sua presença crescente na composição de produtos de uso específico, tais como automóveis, eletrodomésticos e dispositivos de comunicação em geral, torna-se necessário planejar detalhadamente cada etapa do projeto de fabricação, feita de forma sequencial e dividida nas etapas de projeto lógico e projeto físico.

Na etapa de projeto físico, os módulos do projeto são representados geometricamente de forma que, quando alocados nas camadas de materiais correspondentes, garantirão o funcionamento necessário do circuito. O problema de planejamento de *layout* de um VLSI, chamado de *floorplanning*, é um grande desafio na fabricação de um circuito e tem impacto no desempenho, área utilizada, confiabilidade, eficiência e custo do projeto. O *floorplanning* consiste no processo de determinar a localização física dos módulos (itens) e interconectá-los dentro dos limites do chip, utilizando os recursos disponíveis da melhor forma possível, e pode ser interpretado como um problema de empacotamento com retângulos flexíveis.

Os problemas de empacotamento são estudados há décadas como uma ferramenta de apoio na tomada de decisão em diversos setores. Com uma forte característica geométrica, os problemas de empacotamento bidimensionais consistem em alocar itens retangulares menores, com dimensões possivelmente variáveis, em regiões retangulares maiores, utilizando os recursos disponíveis da melhor forma possível e sem que haja sobreposição. Computacionalmente, ele é um problema NP-difícil. Kahng et al. [2011].

Nos circuitos integrados, os itens retangulares são módulos que devem ser alocados no chip e conectados por um fio, sendo o 'comprimento de fio' a distância total utilizada para conectar os módulos pertencentes a uma mesma rede, e cada rede um subconjunto de módulos interconectados. Para minimizar o comprimento do fio usado nas interconexões do circuito, é preciso identificar quais itens devem ser colocados próximos uns aos outros e alocá-los de forma a atender metas, muitas vezes conflitantes, de comprimento de fio utilizado, espaço disponível no chip e desempenho necessário do circuito.

Devido ao grande número de variáveis envolvidas no problema, o *floorplanning* é difícil de ser solucionado, apesar de ser representado facilmente através de modelos matemáticos. Consequentemente, obter soluções exatas torna-se inviável, principalmente para grandes instâncias. Desta forma, métodos heurísticos podem ser eficientes para encontrar boas soluções em tempos computacionais viáveis. Para resolver o problema de empacotamento de retângulos flexíveis em circuitos VLSI, obtendo boas soluções com tempo computacional inferior em relação às abordagens analíticas exatas, heurísticas baseadas em modelos da literatura serão exploradas.

Neste trabalho, o *floorplanning* é abordado como um problema de empacotamento de retângulos flexíveis. Um modelo matemático e uma matheurística de janela deslizante são propostos para resolver de maneira eficiente o problema de planejamento de circuitos VLSI. O procedimento proposto explora simultaneamente as vantagens da programação matemática e do método heurístico, através de uma estratégia de inicialização e da resolução iterativa de um modelo com parte dos módulos retangulares do problema original. Ao incluir a janela deslizante, uma quantidade fixa de variáveis livres são consideradas, diminuindo significativamente a dificuldade computacional do modelo. A influência de diferentes tamanhos de janela deslizante e critérios na ordenação inicial sobre o resultado final será analisada.

2. Revisão de Literatura

Nesta seção é apresentada uma breve revisão de literatura sobre métodos para resolver o problema de empacotamento de retângulos flexíveis, genericamente abordados, além de métodos aplicados no contexto específico de planejamento de *layout* de circuitos VLSI.

Murata e Kuh [1998] propuseram um método para resolver o problema de empacotamento de um conjunto de módulos mistos flexíveis, baseado na representação por pares de sequências (*sequence pair*). Um par de sequência é um par de permutações entre dois retângulos e especifica qual das condições de posição relativa é válida, sendo elas atribuídas aos módulos como inequações. Uma formulação para o problema foi proposta e o problema resolvido usando o método de pontos interiores com uma instância de 49 módulos.

Para resolver o problema de empacotamento de retângulos, Young et al. [2001] utilizou a relaxação Lagrangiana, implementada em um *simulated annealing framework* e com objetivo de minimização da área total obtida no empacotamento dos módulos. Também com foco em minimizar a área de ocupação, Wu et al. [2016] propôs um procedimento heurístico para resolver o problema quando o *layout* final deveria conter um retângulo central, com flexibilidade nas dimensões, que poderiam ser alteradas dentro de limites razoáveis pré-estabelecidos. Estratégias de preenchimento de espaço interno foram propostas para melhorar a taxa de preenchimento do *layout* final.

No contexto específico do *floorplanning*, com uma abordagem semelhante aos autores Young et al. [2001], Chi e Chi [2002] propuseram um algoritmo com módulos flexíveis para otimizar a área utilizada no projeto de um circuito VLSI. Ibaraki e Nakamura [2006] abordaram o problema de empacotamento de retângulos utilizando pares de sequência para especificar posições relativas entre cada par de retângulos. Para determinar o tamanho exato e a localização de todos os retângulos, uma formulação matemática foi proposta e para encontrar bons pares de sequências, técnicas de busca local foram utilizadas.

Por sua vez, Chen e Chang [2009] propuseram uma abordagem analítica exata tratando as posições relativas dos módulos retangulares como variáveis binárias do problema no *floorplanning*. Para minimizar a área de forma linear, a largura do retângulo envolvente foi fixada e a altura necessária para a alocação dos módulos minimizada na função objetivo, respeitando limites pré-definidos. O modelo matemático formalizado pelos autores apresentou resultados satisfatórios, podendo considerar módulos flexíveis e gerar soluções exatas. Apesar disso, a formulação possui complexidade computacional de um problema linear inteiro misto e seu tempo de execução se torna bastante penoso para médias e grandes instâncias.

Laskar et al. [2015] apresentou um estudo e comparação dos diferentes algoritmos de otimização e as representações envolvidas no problema de *layout* VLSI. Entre os algoritmos, a maior parte do trabalho foi realizada para otimizar a minimização da área. Singh et al. [2016] reuniu e revisou heurísticas e algoritmos meta-heurísticos de trabalhos que consideram o problema de otimização no planejamento VLSI. Comparações entre os procedimentos montados foram incluídas neste estudo e mesmo a representação mais efetiva apresentada indicou limitações de flexibilidade.

Apesar dos diversos trabalhos abordando o problema de empacotamento de retângulos, principalmente no contexto do *floorplanning* de módulos flexíveis em circuitos VLSI, métodos de solução mais eficientes podem ser explorados, garantindo que muitos estudos ainda sejam desenvolvidos e possibilitando o uso de instâncias mais próximas do ponto de vista prático. Para resolver este problema, um método exato, inspirado em Chen e Chang [2009], e uma formulação heurística serão apresentados neste trabalho.

3. Fundamentação Teórica

Nesta seção, conceitos essenciais para a compreensão da abordagem proposta neste trabalho são apresentados.

3.1. Representação de Posicionamento

Para representar a configuração geométrica dos módulos no *layout*, as posições dos itens podem ser especificadas por coordenadas definidas em relação a um sistema de eixos ou através de um conjunto de relações topológicas entre os itens, definindo as posições relativas entre eles. Neste trabalho, foi utilizada uma representação derivada da ferramenta padrão, conhecida como par de sequência [Murata et al., 1996], para especificar a posição relativa entre cada par de módulos e obter um empacotamento sem sobreposição.

Assim como em Chen e Chang [2009], dado dois módulos i e j , as variáveis binárias de posicionamento p_{ij} e q_{ij} , horizontal e vertical, respectivamente, são introduzidas para denotar a posição que o item i ocupa em relação ao item j . Além disso, p_{ji} e q_{ji} complementam a posição dos itens i e j . Apenas uma das variáveis binárias associadas ao par de itens assume valor não nulo, assim, os módulos i e j são considerados não sobrepostos se uma das condições apresentadas na Tabela 1 é válida.

Tabela 1: Valores das variáveis binárias ao posicionar dois módulos i e j

	p_{ij}	q_{ij}	p_{ji}	q_{ji}
i está acima de j	0	0	0	1
i está à direita de j	0	0	1	0
i está abaixo de j	0	1	0	0
i está à esquerda de j	1	0	0	0

3.2. Cálculo do Comprimento de Fio

Nos circuitos integrados, uma rede r é um conjunto de módulos (itens retangulares) interconectados. Para calcular o comprimento do fio gasto para fazer essas conexões, recorre-se à aproximação, dado que as informações reais de roteamento não estão disponíveis no estágio do *floorplanning*. Uma aproximação de comprimento de fio comumente usada para uma rede é medir seu comprimento de fio de meio-perímetro ponderado (HPWL). O método HPWL é razoavelmente preciso, calculado com eficiência e adota como distância o comprimento de meio perímetro da menor caixa delimitadora (retângulo) que inclui todos os pontos de conexão do circuito integrado, também chamados de pinos. Quando as posições dos pinos não são fornecidas, pode-se calcular o HPWL usando os centros dos módulos flexíveis [Chen e Chang, 2009; Kahng et al., 2011].

Neste trabalho, o cálculo do HPWL é obtido a partir do comprimento de meio perímetro do retângulo exterior cuja arestas contém o centro dos módulos que estão ligados entre si, isto é, a distância entre os dois pontos centrais dos módulos é dada pela soma das diferenças absolutas de suas coordenadas, utilizando apenas retas verticais e horizontais. Este formato não-euclidiano de obter distâncias d_x e d_y é conhecido como Geometria do Táxi ou Distância de *Manhattan*. O comprimento de fio entre dois itens é dado por $d = d_x + d_y$, sendo:

$$d_x = |x_j - x_i| \quad \forall i, j \in J \quad (1)$$

$$d_y = |y_j - y_i| \quad \forall i, j \in J \quad (2)$$

Para obter as dimensões W_r e H_r , respectivamente, largura e altura de uma rede r , calcula-se a soma da distância central de todos os módulos interconectados. Sabendo que o comprimento de fio, utilizado nas interconexões, tem grande influência nos custos e desempenho de um circuito, o modelo matemático proposto busca minimizar as distâncias associadas ao problema de empacotamento de retângulos flexíveis abordado. Ao adotar uma função objetivo que reduz a distância HPWL do circuito VLSI, espera-se obter *layouts* compactos e eficientes.

4. Modelo Matemático

O problema de posicionamento de módulos flexíveis em circuitos VLSI considera como variáveis de decisão as posições ideais para um determinado conjunto de módulos alocados em uma área de chip, de modo que não haja sobreposição, além de decidir quais as dimensões mais adequadas para cada retângulo flexível. Tem como objetivo minimizar as distâncias do comprimento de fio HPWL entre subconjuntos especificados dos itens, isto é, reduzir a medida da quantidade de metal necessária para interconectar os módulos de forma a garantir a funcionalidade do chip. Os parâmetros e variáveis do modelo matemático proposto são definidos por:

Parâmetros:

- n : número de módulos flexíveis;
- p : número de redes;
- J : conjunto de itens i , $i \in \{1, \dots, n\}$;
- R : conjunto de redes r , $r \in \{1, \dots, p\} \subseteq J$, subconjuntos de J ;
- w_i^{min}, h_i^{min} : dimensões mínimas do item i ;
- w_i^{max}, h_i^{max} : dimensões máximas do item i ;
- s_i^{min}, s_i^{max} : proporção mínima e máxima entre altura e largura do item i (*aspect ratio*);
- W^{max}, H^{max} : dimensões máximas do retângulo envolvente (largura e altura do chip).

Variáveis:

- x_i : abcissa do item i (variável de posição);
- y_i : ordenada do item i (variável de posição);
- w_i, h_i : dimensões do item i ;
- W_r, H_r : dimensões da rede r ;
- p_{ij} e q_{ij} : relação horizontal e vertical de posição, respectivamente;

$$p_{ij} = \begin{cases} 1, & \text{se } i \text{ está à esquerda de } j; \\ 0, & \text{caso contrário.} \end{cases} \quad q_{ij} = \begin{cases} 1, & \text{se } i \text{ está abaixo de } j; \\ 0, & \text{caso contrário.} \end{cases}$$

Modelo:

minimize

$$\sum_{r \in R} W_r + H_r \quad (3)$$

sujeito a :

$$w_i^{\min} \leq w_i \leq w_i^{\max} \quad \forall i \in J \quad (4)$$

$$h_i^{\min} \leq h_i \leq h_i^{\max} \quad \forall i \in J \quad (5)$$

$$s_i^{\min} \cdot h_i \leq w_i \leq s_i^{\max} \cdot h_i \quad \forall i \in J \quad (6)$$

$$x_j \geq x_i + w_i - (1 - p_{ij}) \cdot W^{\max} \quad \forall i \neq j \in J \quad (7)$$

$$y_j \geq y_i + h_i - (1 - q_{ij}) \cdot H^{\max} \quad \forall i \neq j \in J \quad (8)$$

$$p_{ij} + q_{ij} + p_{ji} + q_{ji} = 1 \quad \forall i < j \in J \quad (9)$$

$$W_r \geq |(x_i + \frac{w_i}{2}) - (x_j + \frac{w_j}{2})| \quad \forall r \in R, \forall i, j \in r \quad (10)$$

$$H_r \geq |(y_i + \frac{h_i}{2}) - (y_j + \frac{h_j}{2})| \quad \forall r \in R, \forall i, j \in r \quad (11)$$

$$p_{ij}, q_{ij} \in \{0, 1\} \quad \forall i \neq j \in J \quad (12)$$

No modelo (3)-(12), a função objetivo (3) minimiza o comprimento de fio utilizado nas interconexões dos itens, de todas redes, adotando a simplificação de distância entre dois módulos que, neste caso, é sempre calculada entre os centros geométricos dos módulos. As restrições (4) e (5) garantem que as dimensões dos módulos flexíveis estejam dentro do intervalo permitido. A restrição (6) estabelece que cada módulo retangular deve respeitar uma proporção altura-largura ($s_i = h_i/w_i$) para obter um *layout* eficiente.

As restrições (7) e (8), referentes às relações de posicionamento entre os itens, garantem que não haja sobreposição entre dois módulos e que ambos estejam contidos no retângulo envolvente do *layout*, isto é, estejam inteiramente alocados na região do chip. A restrição (9) assegura que uma, e apenas uma, das variáveis binárias de posicionamento referentes a i e j assumam valor não nulo. As restrições (10) e (11) são referentes às dimensões de rede, calculadas, rede a rede, para todos os itens contidos nela. A restrição (12) define p_{ij} e q_{ij} o domínio das variáveis de decisão.

5. Matheurística

O problema de empacotamento de retângulos flexíveis pode ser facilmente representado através de modelos matemáticos, porém é difícil de ser solucionado devido ao grande número de variáveis envolvidas. Obter soluções exatas implica em alta complexidade computacional, principalmente para problemas com grandes dimensões. Para explorar o espaço de soluções com um baixo esforço computacional e garantia de boas soluções, métodos heurísticos são frequentemente utilizados [Singh et al., 2016; Wu et al., 2016]. Assim, uma abordagem matheurística de janela deslizante é proposta nesta seção para resolver de maneira eficiente o problema, aplicado no contexto do *floorplanning* de circuitos VLSI.

Combinando programação matemática e um método heurístico para reduzir a complexidade do problema, a matheurística proposta é um procedimento iterativo que conta com uma estratégia de inicialização, estabelecida utilizando critérios de ordenação para os itens que se deseja alocar. Desta forma, é possível definir quais módulos serão priorizados no planejamento do circuito e resolver um problema parcial, com apenas alguns dos módulos a serem alocados, ao invés de buscar uma solução exata para o problema. A cada iteração, um novo item não-alocado é selecionado

para ser inserido no *layout* e um sub-modelo (3)-(12) é criado com tamanho fixo e reduzido em relação ao modelo do problema original.

Para formalizar a matheurística, novos parâmetros foram adotados:

- sw : tamanho da janela deslizante, $sw \in \{1, \dots, n - 1\}$;
- con_i : conectividade do item i , $con_i \in \{0, \dots, p\}$;
- it : índice de iteração, $it \in \{0, \dots, n - sw\}$.

O tamanho da janela deslizante sw indica a quantidade de variáveis livres (variáveis de decisão) consideradas na resolução do sub-modelo, isto é, o número de itens que podem ser posicionados, reposicionados ou ajustados em suas dimensões a cada iteração. O parâmetro sw pode assumir valor entre 1 e $n - 1$, sendo n a quantidade total de itens do problema. A conectividade con_i de um item i indica o número de redes r que o contém, calculado no momento de inicialização do algoritmo e limitado no intervalo de 0 a p , sendo p o número total de redes do problema. Os valores sw e con_i são constantes durante todo o processamento da matheurística.

Na primeira etapa da matheurística, é utilizada uma estratégia de inicialização que consiste em ordenar os módulos flexíveis seguindo um critério de ordenação, dentre os critérios:

- área de forma não-decrescente;
- área de forma não-crescente;
- conectividade de forma não-decrescente;
- conectividade de forma não-crescente.

Na etapa seguinte, um sub-modelo é criado para resolver a alocação de parte dos itens do problema original na primeira iteração ($it = 0$). Este sub-modelo é criado utilizando (3)-(12). Apenas os sw primeiros itens da lista ordenada são considerados na resolução, diminuindo significativamente a complexidade computacional do problema. O sub-modelo é resolvido e, a partir da sub-solução obtida, fixam-se as posições relativas do primeiro item da alocação feita, isto é, as variáveis associadas ao item passam a ser parâmetros no sub-modelo. Após acrescentar o próximo item não-alocado da lista ordenada ao sub-modelo, inicia-se uma nova iteração ($it = 1$). O sub-modelo é resolvido e, a partir da sub-solução obtida, fixam-se as posições relativas do segundo item da lista ordenada.

O sub-modelo é atualizado, acrescentando o terceiro item não-alocado, e resolvido. A partir da sub-solução obtida, fixam-se as posições relativas referentes ao item $it + 1$, sendo it o índice da iteração presente, e atualiza-se o sub-modelo inserindo o próximo item não-alocado da lista. O sub-modelo é resolvido novamente. Estes passos são realizados de modo iterativo até que a quantidade de itens do sub-modelo seja igual à quantidade de itens do modelo inicial. Adota-se como solução do modelo a solução obtida para o sub-modelo na última iteração realizada.

6. Resultados

Para avaliar o desempenho do procedimento proposto, testes computacionais foram realizados na linguagem de programação C++ com o solver CPLEX e executados em um computador com processador Intel Core i7-3517U com 8GB de memória RAM. Foram utilizadas cinco instâncias de referência arquivadas no *Microelectronics Center of North Carolina (MCNC)* com origem no projeto de circuitos VLSI, nas quais todos os itens têm forma retangular flexível. Cada instância contém um número de redes, sendo cada rede um conjunto de módulos (itens) interconectados. São elas APTE, XEROX, HP, AMI33 e AMI49.

As instâncias foram testadas para três diferentes valores do parâmetro sw , que corresponde ao tamanho da janela deslizante no modelo e indica a quantidade de variáveis livres a serem consideradas por iteração, variando nos valores $sw = [1, 3, 5]$, sendo 1 o menor valor possível para sw , 3 e 5 valores escolhidos arbitrariamente. Ainda, foi estipulado o limite de tempo de 300 segundos para a obtenção de cada solução por iteração.

A matheurística tem como resultado as posições ideais para o conjunto de itens alocados, suas dimensões, o comprimento de fio HPWL utilizado para interconectá-los de acordo com as redes e o tempo computacional do procedimento. Através dos resultados obtidos, uma solução geométrica pode ser ilustrada para cada alocação obtida pela abordagem. Na Figura 1, por exemplo, é possível visualizar a alocação gerada para a instância AMI49 com $sw = 3$ e ordenação pela área de modo não-decrescente.

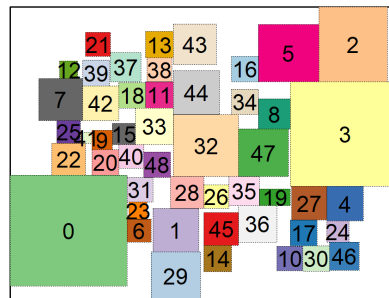


Figura 1: Alocação obtida para AMI49: itens ordenados pela área não-decrescente ($sw = 3$).

Os valores obtidos pelo modelo matemático (3-12) exato, de forma isolada sem agregar a matheurística, com tempo de execução limitado a 54000 segundos, foram rapidamente superados pelos resultados encontrados pela matheurística. Para as instâncias AMI33 e HP, a função objetivo do modelo exato nessa condição de tempo foi de 24610 e 33194, respectivamente. Esses valores indicam que através da abordagem matheurística proposta é possível atingir boas soluções em um tempo muito menor.

Nas tabelas 2 a 6 são apresentados os valores de comprimento de fio e tempo computacional, em segundos, obtidos pela matheurística para cada instância, combinando diferentes critérios de ordenação inicial e tamanhos de janela. Destaca-se o melhor resultado obtido para a instância.

Na Tabela 5, observa-se que a melhor solução encontrada para a instância HP foi obtida ao adotar o critério de ordenação por conectividade de modo não-crescente, considerando uma janela deslizando de tamanho 5. Para a instância AMI49, o melhor resultado obtido considerou como o critério de inicialização a conectividade de itens modo não-decrescente, característica que se manteve para as demais instâncias: AMI33, APTE e XEROX.

Ao definir um critério de ordenação inicial é possível estabelecer quais itens serão priorizados na alocação feita e, com isso, melhorar o processo iterativo realizado. De acordo com

	$sw = 1$		$sw = 3$		$sw = 5$	
AMI33	Solução	Tempo	Solução	Tempo	Solução	Tempo
Menor área (a)	26862,5	97,797	24157	6496,74	23786	22632,5
Maior área (b)	28546	30,969	24437	2936,82	25928	16784,5
Menor conectividade (c)	16492	121,361	15393	11503,3	15515,5	20845
Maior conectividade (d)	23590	148,549	21189	8349,35	22606,5	14821,7

Tabela 2: Comprimento de fio e tempo computacional por ordenação para a instância AMI33.

	$sw = 1$		$sw = 3$		$sw = 5$	
AMI49	Solução	Tempo	Solução	Tempo	Solução	Tempo
Menor área (a)	356814	1055,4	436484	11056,4	464532	10982,6
Maior área (b)	403112	191,598	421536	580,204	402972	3184,63
Menor conectividade (c)	344452	246,361	308557	3108,19	306202	12574,9
Maior conectividade (d)	362426	409,797	437191	4189,91	355374	7201,85

Tabela 3: Comprimento de fio e tempo computacional por ordenação para a instância AMI49.

	$sw = 1$		$sw = 3$		$sw = 5$	
APTE	Solução	Tempo	Solução	Tempo	Solução	Tempo
Menor área (a)	130074	4,893	124739	47,281	116978	137,865
Maior área (b)	197494	4,627	126250	32,127	126250	435,431
Menor conectividade (c)	122450	4,559	108274	28,729	104486	641,047
Maior conectividade (d)	165630	6,541	161028	26,559	159086	743,925

Tabela 4: Comprimento de fio e tempo computacional por ordenação para a instância APTE.

	$sw = 1$		$sw = 3$		$sw = 5$	
HP	Solução	Tempo	Solução	Tempo	Solução	Tempo
Menor área (a)	61299	4,333	50925	56,88	44643	999,131
Maior área (b)	47090	3,638	34979,5	37,969	34395	914,076
Menor conectividade (c)	49349	4,999	41198,5	45,393	38448,5	232,798
Maior conectividade (d)	41378	3,611	32529	98,133	31203	1013,09

Tabela 5: Comprimento de fio e tempo computacional por ordenação para a instância HP.

	$sw = 1$		$sw = 3$		$sw = 5$	
XEROX	Solução	Tempo	Solução	Tempo	Solução	Tempo
Menor área (a)	413807	5,689	410370	23,386	381810	628,338
Maior área (b)	413008	4,837	321813	8,191	323108	687,003
Menor conectividade (c)	358890	5,461	315004	16,304	327607	274,853
Maior conectividade (d)	528906	5,804	448472	9,282	390386	170,678

Tabela 6: Comprimento de fio e tempo computacional por ordenação para a instância XEROX.

os dados supracitados nas tabelas 2 a 6, para todas as instâncias testadas, os melhores resultados para os testes foram obtidos ao utilizar, como estratégia, ordenações baseadas nas características da descrição detalhada da conectividade do VLSI.

As soluções obtidas indicam grande influência do critério de ordenação no resultado final do problema. Com tendência a obter o melhor resultado, ordenar os itens pela conectividade de modo não-decrescente garante que o processo de alocação seja iniciado fixando primeiramente os itens que possuem menor conectividade e flexibilizando o posicionamento dos itens que aparecem em um número maior de conexões. Na figura 2 é possível visualizar a solução geométrica (*layouts*) de algumas alocações obtidas para a instância AMI33 com diferentes tamanhos de janela e ordenações por conectividade.

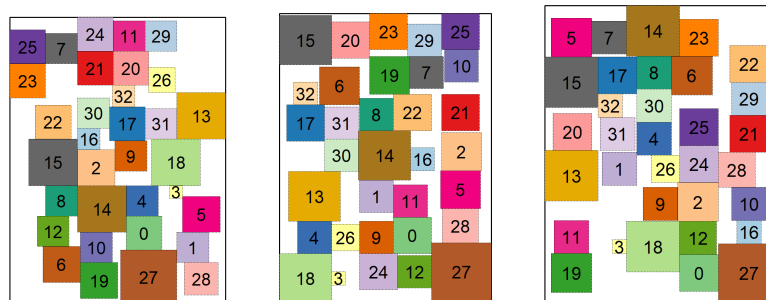


Figura 2: Alocações obtidas para AMI33: itens ordenados por maior conectividade ($sw = 1$, $sw = 3$ e $sw = 5$, respectivamente).

Na Figura 3 é apresentada graficamente a comparação de soluções entre as instâncias para cada critério de ordenação. No eixo vertical, é expresso o percentual que indica a distância da solução em relação a melhor solução obtida para aquela instância. No eixo horizontal, cada instância é indicada pelo seu nome seguindo, respectivamente, o tamanho de janela sw igual a 1, 3 e 5. O critério que prioriza itens com menor conectividade fica em destaque, com valores inferiores aos demais critérios nas instâncias AMI33, AMI49, APT e XEROX.

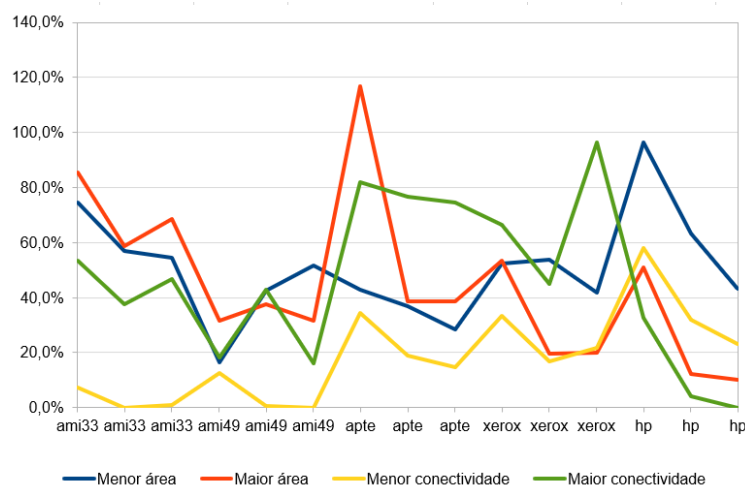


Figura 3: GAP de soluções entre instâncias por critério de ordenação.

Ainda, vale observar a relação entre o tempo computacional e o tamanho de janela adotado. Como proposto na abordagem, ao criar sub-modelos com uma quantidade limitada de variáveis livres, a complexidade computacional é reduzida e o tempo de execução diminui consideravelmente. Tamanhos de janelas maiores tendem a gerar resultados melhores, porém exigem tempos de processamento muito mais elevados, nem sempre viáveis. Por exemplo, na Tabela 3, para a instância AMI49 com ordenação por conectividade não-decrescente, a solução para janela de tamanho 5 é cerca de 12% melhor do que a solução para janela 3, enquanto o tempo computacional sofre um aumento de 510%.

7. Conclusões

Neste trabalho, o problema de planejamento de *layouts* de circuitos VLSI foi abordado como um problema de empacotamento de retângulos flexíveis. Um modelo matemático e abordagem heurística de janela deslizante foram propostos para a obtenção de soluções para o *floorplanning* com módulos flexíveis. Testes computacionais foram implementados em C++ com o solver CPLEX, realizados com conjuntos de referência e indicam a eficiência do método apresentado.

Utilizando uma estratégia de ordenação inicial e criação de sub-modelos com variáveis livres limitadas, foi possível diminuir significativamente a complexidade computacional do modelo e encontrar soluções satisfatórias para o problema em um tempo computacional muito menor. Os resultados demonstram que bons critérios de inicialização têm grande influência sobre a qualidade da solução e estão intimamente ligados às características de conectividade do circuito. Explorar melhores conjuntos de ordenação inicial pode economizar tempo e processamento durante a execução da abordagem, de forma simples e pouco custosa.

Apesar do desempenho satisfatório da abordagem, a característica de flexibilidade dos itens foi pouco explorada. Como continuidade deste trabalho, espera-se incentivar o ajuste de dimensão dos módulos flexíveis, explorar novos critérios de inicialização e implementar uma estratégia de finalização na busca de lapidar a solução encontrada com possíveis modificações nos itens livres permitidos pela matheurística.

Referências

- Chen, T.-C. e Chang, Y.-W. (2009). Floorplanning. In *Electronic Design Automation*, p. 575–634. Elsevier.
- Chi, J. e Chi, M. (2002). An effective soft module floorplanning algorithm based on sequence pair. *IEEE International ASIC/SOC Conference*, 15.
- Ibaraki, T. e Nakamura, K. (2006). Packing problems with soft rectangles. In *International Workshop on Hybrid Metaheuristics*, p. 13–27. Springer.
- Kahng, A. B., Lienig, J., Markov, I. L., e Hu, J. (2011). Global and detailed placement. In *VLSI Physical Design: From Graph Partitioning to Timing Closure*, p. 95–126. Springer.
- Laskar, N., Sen, R., Paul, P., e Baishnab, K. (2015). A survey on vlsi floorplanning: Its representation and modern approaches of optimization. *IEEE Sponsored 2nd International Conference on Innovations in Information Embedded and Communication Systems*.
- Murata, H., Fujiyoshi, K., Nakatake, S., e Kajitani, Y. (1996). Vlsi module placement based on rectangle-packing by the sequence-pair. *IEEE Transactions on Computer*, p. 1518–1524.

- Murata, H. e Kuh, E. S. (1998). Sequence-pair based placement method for hard/soft/pre-placed modules. In *Proceedings of the 1998 international symposium on Physical design*, p. 167–172. ACM.
- Singh, R., Baghel, A., e Agarwal, A. (2016). A review on vlsi floorplanning optimization using metaheuristic algorithms. *International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, p. 4198–4202.
- Wu, L., Zhang, L., Xiao, W.-S., Liu, Q., Mu, C., e Yang, Y. (2016). A novel heuristic algorithm for two-dimensional rectangle packing area minimization problem with central rectangle. *Computers & Industrial Engineering*, 102:208–218.
- Young, F., Chu, C. C., Luk, W., e Wong, Y. (2001). Handling soft modules in general nonslicing floorplan using lagrangian relaxation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 20(5):687–692.