

Estudo Paramétrico de uma Heurística Construtiva para o Problema da Minimização da Área da Envolvente Rectangular no Posicionamento de Rectângulos

Marisa J. Oliveira

ISEP - Instituto Superior de Engenharia do Porto
Rua Dr. António Bernardino de Almeida, 431 4200-072 Porto Portugal
INESC Porto- Instituto de Engenharia de Sistemas e Computadores do Porto
Campus da FEUP Rua Dr.Roberto Frias,378 - Porto
mjo@isep.ipp.pt

Eduarda Pinto Ferreira

ISEP - Instituto Superior de Engenharia do Porto
Rua Dr. António Bernardino de Almeida, 431 4200-072 Porto Portugal
GECAD - Grupo de Investigação em Engenharia do Conhecimento e Apoio à Decisão
Rua Dr. António Bernardino de Almeida, 431 4200-072 Porto Portugal
epf@isep.ipp.pt

A. Miguel Gomes

FEUP - Faculdade de Engenharia da Universidade do Porto
Rua Dr. Roberto Frias, s/n 4200-465 Porto Portugal
INESC Porto- Instituto de Engenharia de Sistemas e Computadores do Porto
Campus da FEUP Rua Dr.Roberto Frias,378 - Porto
agomes@fe.up.pt

RESUMO

No problema de minimização da área no posicionamento de rectângulos pretende-se posicionar, sem sobreposições, um conjunto de rectângulos de forma a minimizar a envolvente da área rectangular. Este é um problema de optimização combinatoria *NP*-difícil e que pertence à classe de problemas de cortes e empacotamento. É um problema que surge em vários contextos reais tais como o posicionamento de módulos em circuitos *VLSI* e no projecto do *layout* de instalações. Neste artigo propomos uma heurística construtiva para resolver o problema de minimização da área da envolvente rectangular no posicionamento de rectângulos.

PALAVRAS CHAVE. Posicionamento de Rectângulos, Heurística Construtiva, Testes Estatísticos.

ABSTRACT

In the rectangle packing area minimization problem one wishes to pack a set of non-overlapping rectangles while minimizing the enclosing rectangular area. This is a NP-hard combinatorial optimization problem, which belongs to the class of cutting and packing problems. This problem appears in wide range of activities such as module placement in *VLSI* circuits and designing facility layouts. In this paper we propose a constructive heuristic to solve the rectangle packing area minimization problem.

KEYWORDS. Rectangle Packing, Constructive heuristic, Statistical tests.

1. Introdução

No problema de minimização da área no posicionamento de rectângulos (MAPR) pretende-se posicionar, sem sobreposições, um conjunto de rectângulos de forma a minimizar a área da envolvente rectangular. Este é um problema de optimização combinatoria *NP*-difícil que pertence à classe de problemas de corte e empacotamento (C&E), que são problemas combinatorios com uma forte componente geométrica. A componente geométrica dos problemas de C&E está presente no posicionamento dos itens, uma vez que é necessário assegurar que estes não se sobrepõem, enquanto a componente combinatoria surge na selecção dos itens, onde os posicionar, na selecção do objecto grande, etc. Os problemas de C&E podem ser resolvidos por qualquer uma das abordagens e técnicas habituais de resolução de problemas de optimização combinatoria: programação mista, heurísticas, metaheurísticas, etc. Dada a natureza combinatoria destes problemas, as técnicas exactas não são capazes de lidar de forma eficiente com instâncias de grande dimensão tornando-se necessário o uso de heurísticas.

Para resolver o problema MAPR propomos a aplicação de uma heurística construtiva parametrizável, com o objectivo de obter uma solução, de boa qualidade, num baixo tempo computacional. Nesta heurística utilizamos uma estrutura em árvore na representação de padrões guilhotináveis e uma codificação desta estrutura semelhante à *Normalized Polish Expression*¹. A ideia principal desta heurística é unir sucessivamente pares de rectângulos de modo a que a área da envolvente rectangular de todos os rectângulos seja mínima. A abordagem proposta tem dois parâmetros de configuração: a escolha dos rectângulos e a forma de os unir existindo, naturalmente, vários critérios para cada um deles. Foram realizados testes estatísticos para avaliar o desempenho dos vários critérios de cada parâmetro.

Este trabalho está estruturado como se segue: na secção 2 é apresentada uma descrição detalhada do problema; na secção 3 é apresentada a heurística construtiva proposta para resolver o problema MAPR; na 4 secção são apresentados testes computacionais e na secção 5 apresentam-se as principais conclusões.

2. Descrição do Problema da Minimização de Área no Posicionamento de Rectângulos

O objectivo do problema MAPR é minimizar a área da envolvente rectangular que contém todos os rectângulos sem que estes se sobreponham. As dimensões dos rectângulos são dadas, o posicionamento deve ser ortogonal (i.e., com os lados paralelos aos eixos horizontal e vertical) e os rectângulos podem sofrer rotações de 90°. Trata-se de um problema bidimensional e de acordo com a recente tipologia de problemas de C&E apresentada por Wäscher et al. (2007) está classificado como sendo um problema de dimensões variáveis (ODP), pois as dimensões do rectângulo exterior envolvente são variáveis. Leung et al. (1990) mostraram que do ponto de vista da sua resolução o MAPR é um problema *NP*-difícil. É um problema que surge em vários contextos tais como o posicionamento de módulos em circuitos *Very Large Scale Integration* (VLSI) e no projecto de instalações, *Facility Layout* (FL).

Quando se posicionam rectângulos os padrões obtidos podem ser guilhotináveis ou não guilhotináveis. Um padrão diz-se guilhotinável se puder ser obtido por sucessivos cortes horizontais ou verticais, de um lado ao outro do padrão, que o dividem em dois rectângulos, caso contrário diz-se não guilhotinável. Na indústria, padrões guilhotináveis surgem quando o dispositivo de corte opera apenas de forma longitudinal ou paralela aos lados e corta o objecto de uma ponta a outra (exemplo: corte de placas de madeira usando serras circulares). Padrões não guilhotináveis surgem no projecto de circuitos VLSI e de instalações (FL). Os padrões guilhotináveis têm como principal vantagem a rapidez em gerar e avaliar padrões, por outro lado os padrões não guilhotináveis permitem obter soluções mais compactas. Lai et al. (2001)

¹ Normalized Polish Expression - sequência de rectângulos e operações de união onde há uma correspondência 1-1 entre a sequência e o padrão, Wong et al.

mostraram que baseado em compactação 1D (por compactação do padrão na direcção vertical ou na horizontal), os padrões guilhotináveis podem ser transformados em não guilhotináveis.

Uma das decisões fundamentais na representação de um padrão surge na codificação das relações geométricas entre rectângulos, sendo necessário desenvolver representações eficientes, flexíveis e efectivas das relações geométricas entre os rectângulos de um padrão. De seguida apresentam-se as principais representações usadas quer no posicionamento de componentes em circuitos *VLSI* quer no projecto de instalações.

Em relação ao posicionamento de módulos em circuitos *VLSI*, Wong et al. (1986) propuseram um algoritmo para padrões guilhotináveis usando uma estrutura em árvore baseada na *Normalized Polish Expression*. Um marco importante foi o surgimento do *Sequence Pair (SP)* introduzido por Murata et al. (1996) na representação de padrões genéricos. Esta representação especifica posições relativas entre rectângulos, como “à direita de”, “à esquerda de”, “acima de” ou “abaixo de” e não necessita das dimensões dos rectângulos. Imahori et al. (2003) propuseram um algoritmo de descodificação baseado em programação dinâmica para a obtenção de um posicionamento óptimo sob restrições especificadas pela solução codificada. Mais tarde Imahori et al. (2005) propuseram um melhoramento mantendo o *SP* para representar a solução e introduzem técnicas para acelerar a avaliação das soluções em várias vizinhanças. Estas representações surgem inseridas em abordagens baseadas em metaheurísticas.

Relativamente aos problemas de *FL*, Sharma et al. (2006) apresentam um *survey* com as principais abordagens que foram surgido ao longo dos tempos, destacando-se as abordagens baseadas em programação quadrática e em grafos orientados. Tal como é referido em Sharma et al. (2006), a principal desvantagem destas abordagens diz respeito às restrições geométricas, pois só são considerados departamentos com áreas iguais. Além disso, nas abordagens baseadas em grafos as dimensões dos departamentos não são consideradas durante a optimização, apenas são consideradas após o padrão ter sido construído. Estes autores referem que outro tipo de abordagem usado foi a programação inteira mista onde são consideradas restrições para evitar as sobreposições. Para trabalhar com problemas de maior dimensão e ter em consideração restrições mais realistas surgiram outro tipo de abordagens tais como as metaheurísticas, permitindo resolver problemas com maiores dimensões, em tempo computacional considerado razoável.

3. Heurística Construtiva Configurável para o Problema MAPR

Neste trabalho propõe-se uma heurística construtiva para o problema MAPR. Com esta heurística pretende-se obter uma solução, de boa qualidade, num baixo tempo computacional, com o objectivo da solução obtida servir de solução inicial para o desenvolvimento de abordagens baseadas em métodos de pesquisa local. Nesta heurística foi usada uma estrutura em árvore para representar um padrão e uma codificação desta estrutura semelhante à *Normalized Polish Expression* baseada em padrões guilhotináveis.

Nesta abordagem representamos um padrão por uma estrutura em árvore como se mostra na Figura 1. A área sombreada na Figura 1.c) representa o desperdício. Os números na árvore, Figura 1b), identificam os rectângulos e os C_i 's representam o modo de unir dois rectângulos. As folhas da árvore representam os rectângulos iniciais (1,...,5) e cada união dá origem a um novo rectângulo (6, 7 e 8). Esta situação é ilustrada na Figura 1.d), em que a união dos rectângulos 1 e 3 dá origem ao rectângulo 6.

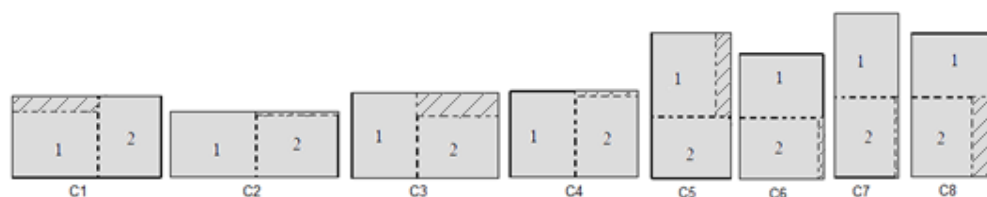
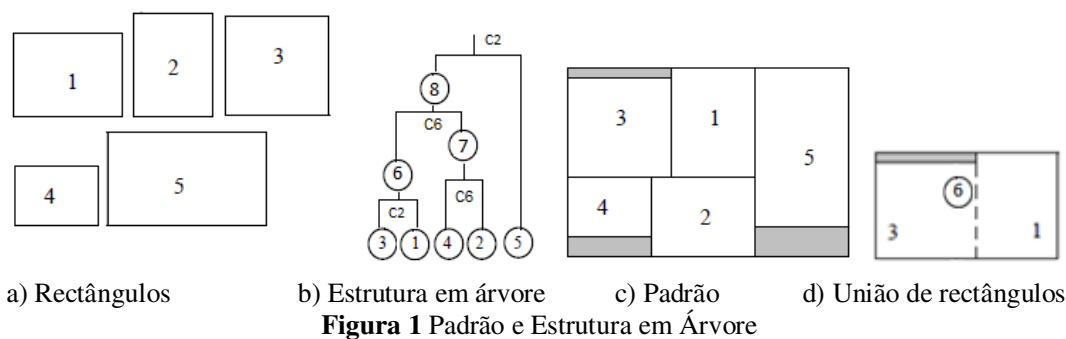


Figura 2 Operações para a união de dois rectângulos

Para a união de dois rectângulos consideramos todas as formas de união possíveis, Figura 2. Podemos observar, na Figura 3, que para esta união, em padrões guilhotináveis, não interessa a posição relativa desses rectângulos, i.e., não interessa qual dos rectângulos está à esquerda ou à direita (acima ou abaixo) uma vez que o rectângulo exterior envolvente é o mesmo.



Figura 3 Posição dos rectângulos no interior do rectângulo envolvente para a operação C_1

Resolver o problema MAPR é equivalente a determinar a estrutura em árvore correspondente ao padrão com valor mínimo da função objectivo. Para representarmos a estrutura em árvore usamos uma representação semelhante à *Normalized Polish Expression* em que é usado um par de sequências em vez de apenas um. Uma das sequências indica a ordem de escolha dos rectângulos e a outra as operações usadas.

Na abordagem proposta a estrutura em árvore da Figura 1.b) é representada pela seguinte codificação (3 1 4 2 6 7 8 5; 2 6 6 2) onde a primeira sequência indica a ordem pela qual os rectângulos foram escolhidos e a segunda qual a operação usada para unir os rectângulos. Por exemplo, os rectângulos 3 e 1 são unidos segundo a operação 2. Na sequência dos rectângulos são também incluídos, para além dos rectângulos iniciais, os rectângulos resultantes de uniões.

A abordagem que usamos tem dois parâmetros de configuração que é necessário escolher, N_1 e N_2 . O parâmetro N_1 é usado para seleccionar o primeiro dos rectângulos a unir. O parâmetro N_2 é usado para seleccionar o segundo rectângulo cuja união com o primeiro resulte num rectângulo que satisfaça o critério seleccionado. O algoritmo inicia com um rectângulo da lista de rectângulos, que satisfaça N_1 . De seguida vai unir este rectângulo com todos os outros

da lista, ainda não usados, e escolhe aquele que o rectângulo resultante da união satisfaça o segundo parâmetro escolhido, N_2 . A união de dois rectângulos é feita segundo uma das operações apresentadas na Figura 2. O rectângulo resultante da união na iteração k , R_{N+k} , é inserido na última posição da lista. O algoritmo termina quando na lista houver apenas um único rectângulo ainda não usado. Este rectângulo é o rectângulo exterior envolvente. De seguida é apresentado um fluxograma com a descrição da heurística construtiva.

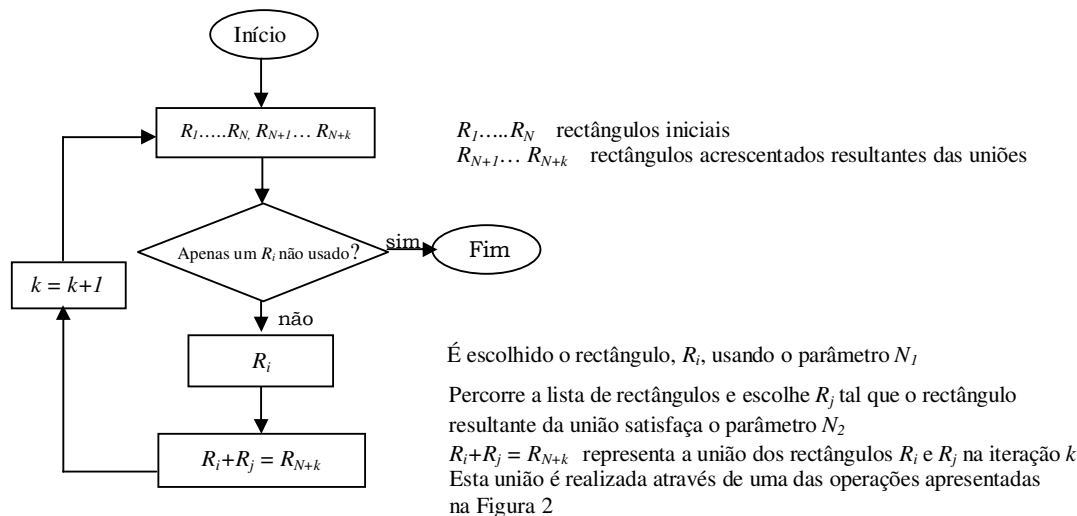


Figura 4 Fluxograma da heurística construtiva

Para parâmetros de configuração N_1 e N_2 foram usados critérios baseados nas escolhas dos rectângulos com maior ou menor largura, altura ou área. Os critérios possíveis são os seguintes:

- E_1 - maior largura;
- E_2 - maior altura;
- E_3 - menor largura;
- E_4 - menor altura;
- E_5 - maior área;
- E_6 - menor área;

4. Testes Computacionais e Estatísticos

Nesta secção apresentamos os resultados computacionais e testes estatísticos realizados para analisar o comportamento da heurística e estudar o efeito dos parâmetros de configuração nos resultados obtidos. Simultaneamente queremos também verificar se os resultados dependem ou não das relações entre as dimensões dos rectângulos (i.e. a forma dos rectângulos) e se dependem do número total de rectângulos considerados. Para esta análise foram usadas várias instâncias cuja construção se descreve na subsecção 4.1. A heurística foi implementada na linguagem C++, utilizando o compilador gcc versão 345, num computador Intel Core i5 2.27 GHz com 4 GB de memória RAM.

4. 1 Instâncias Usadas

Para analisar a dependência das relações entre as dimensões dos rectângulos foram usadas seis classes diferentes de instâncias. As primeiras três classes foram propostas por Martello et al. (1998) e são constituídas por rectângulos de três tipos. Cada tipo consiste na geração de rectângulos com largura e altura (w e h) dados por :

- Tipo 1 - w e h são uniformemente gerados em $\left[\frac{2}{3}W, W\right]$ e $\left[1, \frac{1}{2}H\right]$, respectivamente;
- Tipo 2 - w e h são uniformemente gerados em $\left[1, \frac{1}{2}W\right]$ e $\left[\frac{2}{3}H, H\right]$, respectivamente;
- Tipo 3 - w e h são uniformemente gerados em $\left[\frac{1}{2}W, W\right]$ e $\left[\frac{1}{2}H, H\right]$, respectivamente;

Consideramos $W = H = 100$, com w e h números inteiros. Os rectângulos do tipo 1 têm forma sobre o comprido, os do tipo 2 forma sobre o alto e os do tipo 3 forma sobre o quadrado. As três classes propostas por Martello et al. (1998) são constituídas por rectângulos dos 3 tipos anteriores em diferentes percentagens:

- Classe 1 - rectângulos do tipo 1 gerados com probabilidade de 70% e os restantes com probabilidade 15% de cada um dos outros tipos;
- Classe 2 - rectângulos do tipo 2 gerados com probabilidade de 70% e os restantes com probabilidade 15% de cada um dos outros tipos;
- Classe 3 - rectângulos do tipo 3 gerados com probabilidade de 70% e os restantes com probabilidade 15% de cada um dos outros tipos;

As restantes três classes foram propostas por Berkey et al. (1987), onde w e h são números inteiros, uniformemente gerados em:

- Classe 4 - $[1, 10]$;
- Classe 5 - $[1, 35]$;
- Classe 6 - $[1, 100]$.

A principal diferença entre estas classes e as classes propostas por Martello et al. (1998) é a diversidade de formas. Dentro destas classes a principal distinção está na diversidade dos tamanhos consequência da obrigatoriedade das dimensões serem números inteiros.

Foram realizados testes para estudar se o número total de rectângulos de cada instância tem influência nos resultados obtidos e se há interacção entre os parâmetros escolhidos. Usaram-se instâncias com 20, 50, 100, 250, 500 e 1000 rectângulos. Para cada classe e número total de rectângulos foram geradas 50 instâncias. Para um dado número total de rectângulos, após conjugação dos restantes factores envolvidos, dispõe-se de 10800 valores, resultantes de 50 instâncias $\times 6 N_1 \times 6 N_2 \times 6$ classes. Na totalidade obtiveram-se 64800 valores resultantes de 6×10800 (10800 valores para 20, 50, 100, 250, 500 e 1000 rectângulos). Os resultados obtidos deveriam ser comparados com a solução óptima de cada instância, no entanto, como não é possível obtê-la para todas as instâncias, decidimos comparar com 100% de área usada, ou seja desperdício nulo.

4. 1 Testes Estatísticos

Nesta secção apresentam-se os resultados dos testes estatísticos baseados em comportamentos médios. Para analisar a influência dos parâmetros N_1 e N_2 no valor da FO fez-se uma representação segundo diagramas tipo caixa para se verificar se há opções que são claramente dominadas. Para esta verificação usamos a comparação da mediana com o primeiro quartil, isto é, uma opção domina a outra se a sua mediana é melhor do que o 1º quartil da outra. Simultaneamente pretende-se também verificar se o número total de rectângulos e as classes usadas influenciam os resultados fazendo-se para tal uma análise análoga à anterior.

Na Figura 5 faz-se uma representação do valor da função objectivo (FO) segundo N_1 (5.a)), N_2 (5.b)), o número total de rectângulos (5.c)) e segundo as classes usadas (5.d)).

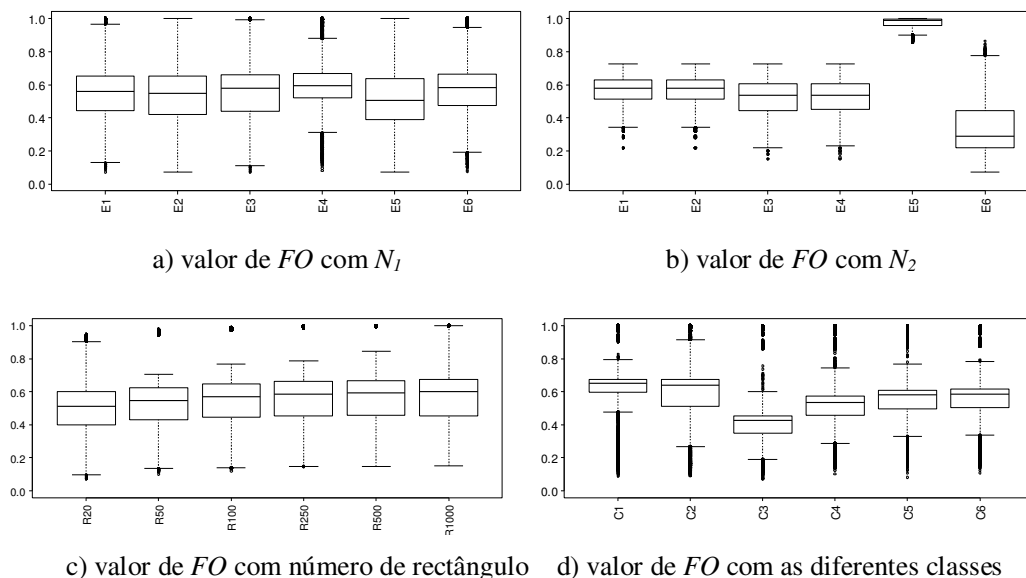


Figura 5 Relacionamento do valor da FO com N_1 , N_2 , número de rectângulos e classes

Para o parâmetro N_1 , número total de rectângulos e classes usadas Figura 5.a), c) e d), respectivamente, não é óbvio por simples visualização dos diagramas tipo caixa que existam comportamentos diferenciados. Já o gráfico da Figura 5.b) sugere comportamentos diferenciados no parâmetro N_2 , em que os piores valores correspondem a E_5 (escolha do rectângulo com maior área). Para este parâmetro verifica-se também a existência de superioridade da opção E_6 (escolha do rectângulo envolvente com menor área) pois a mediana desta opção é melhor do que o primeiro quartil das restantes, levando apenas a considerar N_2 igual a E_6 nos testes que se seguem. Para o parâmetro N_1 constatou-se anteriormente que não era óbvio que existissem comportamentos diferenciados considerando todas as opções do parâmetro N_2 , Figura 5.a). Interessa agora distinguir concretamente, no caso em que para N_2 se considera apenas E_6 se há alguma opção de N_1 que se revele melhor do que as restantes.

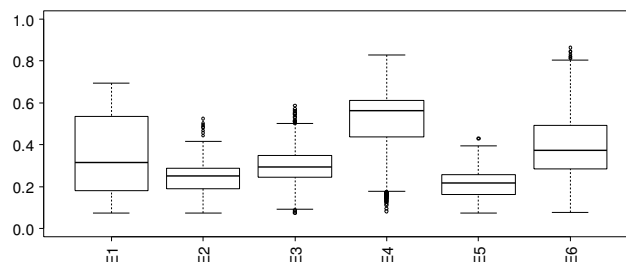


Figura 6 Relaciona o valor da FO com N_1 (para $N_2 = E_6$)

	E_1	E_2	E_3	E_4	E_5	E_6
1º Q	0,18115	1,18802	0,24468	0,43745	0,16443	0,28444
Mediana	0,31503	0,25277	0,29357	0,56126	0,21856	0,37317

Tabela 1 Análise da mediana e 1º quartil das opções de N_1 da Figura 6

A Figura 6 mostra a existência de comportamentos diferenciados no valor da FO , sugerindo uma superioridade dos critérios E_1 (escolha do retângulo com maior largura), E_2 (escolha do retângulo com maior altura) e E_5 (escolha do retângulo com maior área). Estas conclusões podem ser confirmadas através dos valores apresentados na Tabela 1 em que a melhor mediana corresponde à opção E_5 , no entanto, este valor não é melhor do que o primeiro quartil das opções E_1 , E_2 e, obviamente, da própria opção E_5 . Consideraremos apenas estas opções para o parâmetro N_1 e, como já referido anteriormente, a opção E_6 para o parâmetro N_2 nos testes finais.

Para além de avaliarmos a influência dos parâmetros da heurística nos valores da FO pretendia-se também verificar a influência das classes nos resultados. Na Figura 7 apresentam-se diagramas tipo caixa para cada uma das seis classes consideradas para as melhores opções de cada parâmetro acima referidas.

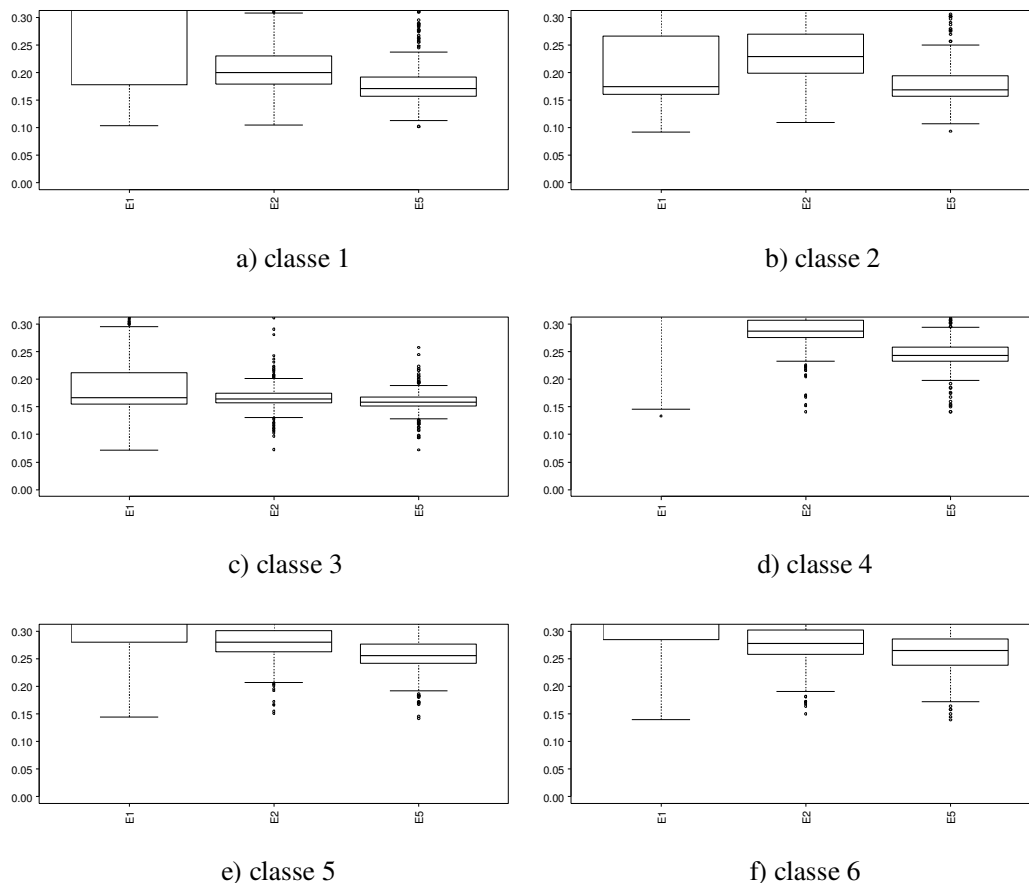


Figura 7 Relaciona o valor da FO com $N_1 = E_1, E_2$ e E_5 e $N_2 = E_6$ para cada classe

Da análise dos diagramas da Figura 7 podemos verificar que a opção E_5 está sempre entre as melhores opções, independentemente da classe de instância, sendo mesmo a melhor opção (opção dominadora) na classe 1 (Figura 7 a)), classe 4 (Figura 7 d)) e classe 5 (Figura 7 e)). Na classe 2 (Figura 7 b)) para além da opção E_5 a outra opção que não é dominada é a opção E_1 . Da Figura 7 c) podemos ainda concluir que, na classe 3, nenhuma das opções é dominante. Finalmente, dos diagramas apresentados conclui-se que na classe 6 (Figura 7 f)) as melhores opções são E_2 e E_5 .

Para avaliar a influência do número de rectângulos nos resultados fez-se uma análise análoga às anteriores. Desagregaram-se os valores por classe de instância para analisar se este número tem influência nas classes usadas e fez-se uma representação do número total de rectângulos segundo diagramas tipo caixa. Estes diagramas apresentam-se na Figura 8. Desta figura podemos concluir que o número total de rectângulos é independente da classe de instância usada.

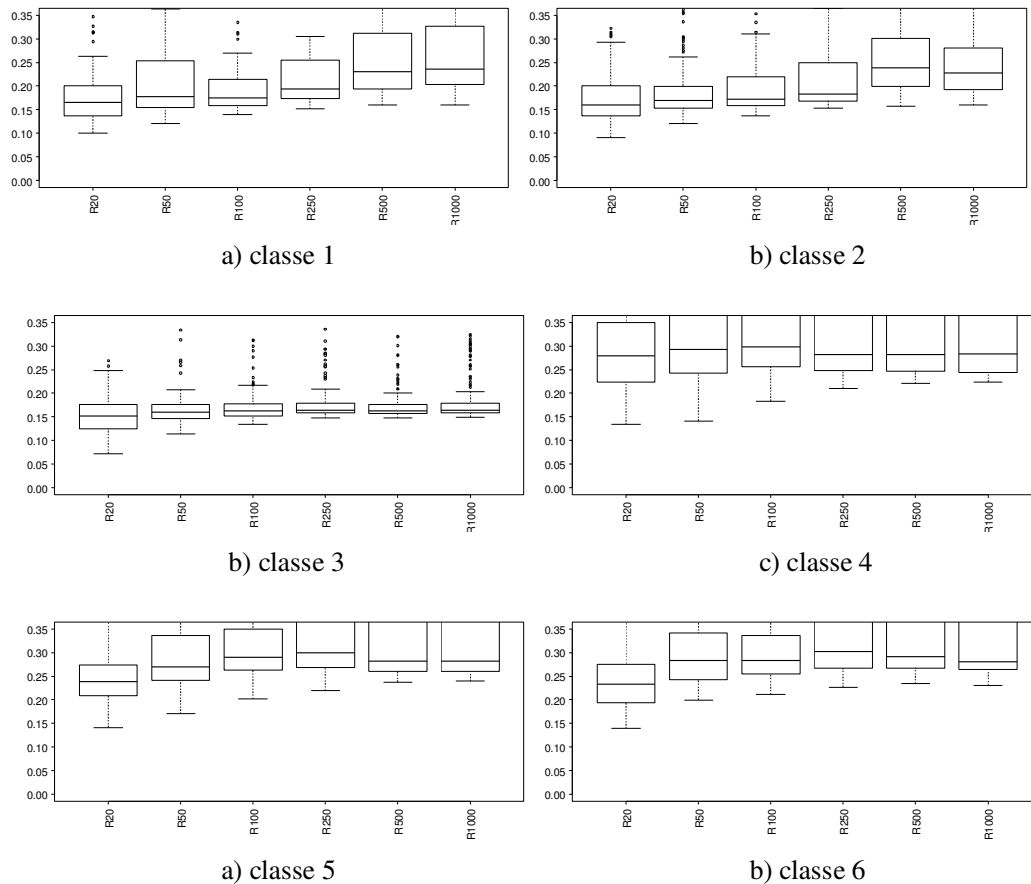


Figura 8 Relaciona o valor da FO com o número total de rectângulos por classe de instância

Finalmente, para estudar se o número total de rectângulos tem influência nos parâmetros usados desagregaram-se os valores por número total de rectângulos representados na Figura 9. Desta figura podemos concluir que não há diferenças significativas entre as opções consideradas

permitindo-nos concluir que o número total de rectângulos não tem influência nas opções dos parâmetros usados.

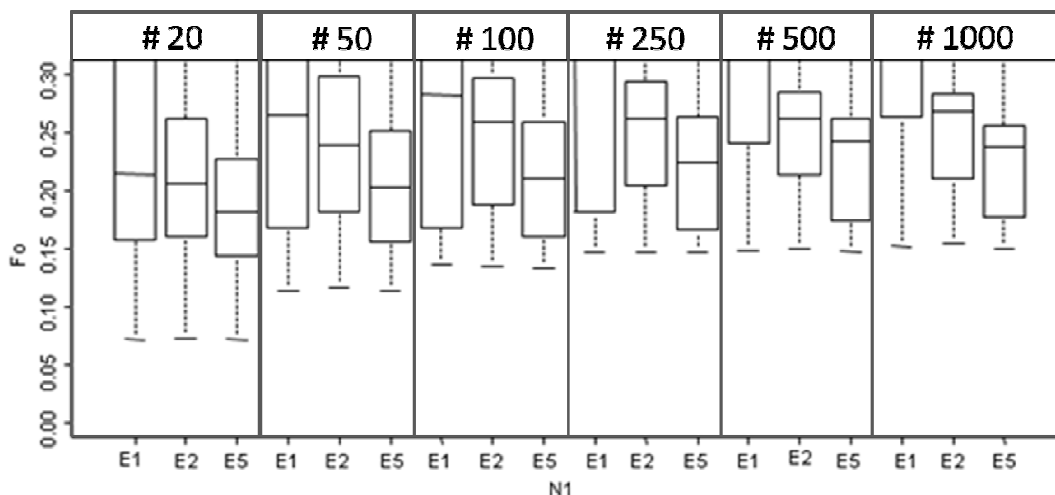


Figura 9 Relaciona o valor da FO com o número total de rectângulos e as opções do parâmetro N_1

5. Conclusões

Neste trabalho desenvolveu-se uma heurística construtiva para o problema bidimensional MAPR. Embora seja um problema fácil de descrever é de difícil resolução. Este problema está classificado como *NP*-difícil o que limita o uso de técnicas exactas para encontrar soluções óptimas em instâncias de grande dimensão. O intuito deste trabalho foi desenvolver uma heurística construtiva para encontrar uma solução, de boa qualidade, num baixo tempo computacional de modo a que a solução obtida possa servir de solução inicial para o desenvolvimento de abordagens baseadas em métodos de pesquisa local.

A heurística construtiva proposta tem dois parâmetros de configuração que é necessário escolher, N_1 e N_2 . Foram realizados testes estatísticos com o objectivo de estudar o efeito desses parâmetros nos resultados obtidos. Estes parâmetros servem para escolher o par de rectângulos a unir e a forma de os unir. N_1 é usado para seleccionar o primeiro rectângulo e N_2 é usado para seleccionar o segundo rectângulo cuja união com o primeiro resulte num rectângulo que satisfaça o critério seleccionado. Foi feita uma comparação entre os diferentes critérios o que nos permitiu identificar os melhores e os piores. Conseguimos limitar as opções para N_2 concluindo que a melhor opção é E_6 (escolha do rectângulo envolvente com menor área). Para o parâmetro N_1 as melhores opções são E_1 (escolha do rectângulo com maior largura) ou E_2 (escolha do rectângulo com maior altura) e E_5 (escolha do rectângulo com maior área).

Os testes estatísticos serviram também para estudar a influência do número total de rectângulos por instância e a influência da relação entre as dimensões dos rectângulos no valor da função objectivo. Os resultados mostraram que, independente da classe usada, a opção E_5 , para o parâmetro N_1 , está sempre entre as melhores opções sendo mesmo a melhor opção quando se considera a classe 1 (rectângulos maioritariamente sobre o comprido), classe 4 (rectângulos pequenos – dimensões entre 1 e 10) e classe 5 (rectângulos médios – dimensões entre 1 e 35). Concluiu-se também que para a classe 3 (rectângulos maioritariamente com forma sobre o quadrado) não há nenhuma opção melhor do que as outras, na classe 2 (rectângulos maioritariamente com forma sobre o alto) as melhores opções são E_1 e E_5 e, finalmente, na

classe 6 (rectângulos diversos – dimensões entre 1 e 100) as melhores opções são E_2 e E_5 . Concluiu-se também que o número total de rectângulos por instância (20, 50, 100, 250, 500 e 1000) não tem influência nos resultados obtidos.

Conseguimos obter uma heurística construtiva parametrizável com três variantes em que a mais genérica é quando se considera E_6 para o parâmetro N_2 com a opção E_5 para o parâmetro N_1 , pois para esta variante os resultados são independentes da classe usada.

A heurística construtiva pode ser usada sozinha ou usar as soluções obtidas como soluções iniciais para o desenvolvimento de abordagens baseadas em pesquisa local. Na primeira situação, para aumentar a robustez dos resultados, uma vez que o tempo computacional para correr cada variante, por instância, é inferior a um segundo, propomos correr consecutivamente as três variantes e escolher o melhor dos três resultados. Quando se pretende usar a solução obtida pela heurística construtiva de modo a servir de solução inicial para o desenvolvimento de abordagens baseadas em métodos de pesquisa local estocásticos podem ser geradas solução de boa qualidade com diferentes características de modo a serem pesquisadas diferentes zonas do espaço de soluções.

6. Referências

- Berkey J.O. and Wang P.Y.** (1987), Two dimensional finite bin packing algorithms, *J. Oper. Res. Soc.*, 38, 423–429.
- Chang Y. C., Chang Y. W., Wu G. M. and Wu S. W.** (2000), B*-trees: A new representation for nonslicing floorplans, *Proc. DAC*, 458-463.
- Chang Y.W and Lin J. M.** (2001), TCG: A Transitive Closure Graph-Based Representation for Non-Slicing Floorplans, *Proc. DAC*, 764-769.
- Chang Y. W and Lin J. M.** (2003), Corner Sequence: A P-Admissible Floorplan Representation With a Worst Case Linear-Time Packing Scheme, *IEEE Trans. on VLSI Syst.*, 4, 679-686.
- Drira A., Pierreval H. and Hajri-Gabouj S.** (2007), Facility layout problems: a survey, *Annual Reviews in Control*, 31,255-267.
- Guo P. N., Cheng C. K. and Yoshimura T.** (1999), An O-tree Representation of Non-Slicing Floorplan and Its Applications. *Proc. DAC*, 268-273.
- Hong X., Huang G., Cai Y., Dong S., Cheng C. K. and Gu J.** (2000), Corner Block List: An effective and efficient topological representation of non-slicing floorplan, *Proc. ICCAD*, 8-12.
- Imahori S., Yagiura M. and Ibaraki T.** (2005), Improved Local Search Algorithms for the Rectangle Packing Problem with General Spatial Costs, *European Journal of Operational Research*, 167, 48-67.
- Imahori S., Yagiura M. and Ibaraki T.** (2003), Local Search Algorithms for the Rectangle Packing Problem with General Spatial Costs, *Mathematical Programming*, 97, 543-569.
- Lai M., Wong D. F.** (2001), Slicing Tree Is a Complete Floorplan Representation, *DATE*, 228-232.
- Leung J., Tam T., Wong C. S., Young G. and Chin F.** (1990), Packing squares into square, *J. Parallel Distrib. Comput.*, 10, 271-275.
- Liu C. L and Wong D. F.** (1986), A new algorithm for floorplan design, *Proc. DAC*, 101-107.
- Martello S. and Vigo D.** (1998), Exact solution of the two-dimensional finite bin packing problem, *Management Science*, 44, 388–399.
- Murata H., Fujiyoshi K., Nakatake S. and Kajitani Y.** (1996), VLSI module placement based on rectangle-packing by the sequence-pair, *IEEE Trans. Comput. Aided Des.*, 15(12),1518-1524.
- Nakatake S., Fujiyoshi K., Murata H. and Kajitani Y.** (1996), Module placement on BSG-structure and IC layout applications, *Proc. Intl. Conf. Comput. Aided Des.*, 484-491.
- Otten R. H. J. M.** (1982), Automatic floorplan design. *Proc. DAC*, 261-267.

- Sharma R. K. and Singh S. P.** (2006), A review of different approaches to the facility layout problems, *Int J Manuf Technol*, 30, 425-433.
- Wäscher G., Haußner H. and Schumann H.** (2007), An Improved Typology of Cutting and Packing Problems, *European Journal of Operational Research*, 183, 1109-1130.
- Wong D. F. and Liu C. L.** (1986), A new algorithm for floorplan design, *Proc. ACM/IEEE Design Automation conference*, 101-107.
- Wong D. F. and Tang X.** (2001), FAST-SP: a fast algorithm for block placement based on sequence pair, *Proc. on Asia P. Des. Autom.*, 521-526.