

Algoritmo Genético para o Problema de Empacotamento 2D Clássico

Anderson Zudio¹, Luiz Satoru Ochi²

Universidade Federal Fluminense

Instituto de Computação – IC/UFF – Av. Gal. Milton Tavares de Souza, s/nº Campus da Praia

Vermelha - Boa Viagem - Niterói / RJ

azudio@id.uff.br¹, satoru@ic.uff.br²

Paulo E. D. Pinto³, Igor M. Coelho⁴

Universidade do Estado do Rio de Janeiro

Instituto de Matemática e Estatística – IME/UERJ – Rua São Francisco Xavier, 524 Sexto Andar -

Maracanã - Rio de Janeiro/RJ

pauloedp@ime.uerj.br³, igor.machado@ime.uerj.br⁴

RESUMO

O problema de empacotamento consiste em empacotar ortogonalmente um conjunto de itens na menor quantidade de caixas possível sem sobreposição. O caso bidimensional generaliza o problema de empacotamento unidimensional clássico amplamente difundido na literatura, portanto, é da classe NP-Difícil. Este trabalho apresenta um algoritmo genético de chaves aleatórias viciadas (BRKGA) para o problema de empacotamento bidimensional, considerando que os itens podem ser alocados com rotação ou orientação fixa. O problema tem várias aplicações industriais, relacionando-se a outros problemas complexos. O algoritmo proposto usa uma estratégia chamada *Variable Cross Descent* (VCD), que acelera o processo evolutivo do BRKGA. O resultado computacional realizado com 500 instâncias da literatura demonstram que o método proposto, denominado BRKGA/VCD, é capaz de obter soluções de qualidade superior ou equivalente aos algoritmos estado da arte encontrados na literatura.

PALAVRAS CHAVE. Meta-heurística. Empacotamento bidimensional. BRKGA.

Tópicos – Meta-heurísticas. Inteligência Computacional.

ABSTRACT

The bin packing problem consists of orthogonally packing a set of boxes into the minimum number of bins without overlap. The two-dimensional bin packing problem generalizes the well known unidimensional bin packing problem, therefore, this problem is NP-Hard. This paper proposes a biased random keys genetic algorithm (BRKGA) for the two-dimensional bin packing problem. The boxes may be packed with fixed or non-fixed orientation. The bin packing problem has many industrial applications and relates to other complex problems. The proposed algorithm uses a novel method called *Variable Cross Descent* (VCD) that accelerates the evolving process of BRKGA. An extensive computational test using 500 literature instances shows that the results devised by the proposed algorithm are equivalent or better in terms of solution quality when compared to state-of-art algorithms found in the literature.

KEYWORDS. Metaheuristics. Two-dimensional Bin Packing. BRKGA.

Paper Topics – Metaheuristics. Computational Intelligence.

1. Introdução

O problema de empacotamento clássico bidimensional (2BP) consiste em empacotar ortogonalmente um conjunto de n itens em forma de retângulos caracterizados por sua altura h_i e largura w_i , $i \in \{1, \dots, n\}$, no menor número possível de caixas homogêneas de altura $H \geq h_i$ e largura $W \geq w_i$. A quantidade de caixas é ilimitada e os itens não podem ser alocados com sobreposição. Deste modo, nenhuma outra restrição é imposta nesta versão do problema. O caso em que os itens têm orientação fixa são estudados separadamente do caso em que há rotação. De acordo com a topologia proposta por Wäscher et al. [2007], o 2BP é classificado como *Single Stock-Size Cutting Stock Problem* (SSSCSP) ou *2D-Single Bin-Size Bin Packing Problem* (2D-SBSBPP). Quando consideramos que todo item i tem sua altura $h_i = H$, o 2BP passa a generalizar o problema de empacotamento clássico unidimensional (1BP ou 1D-SBSBPP). Logo, tais problemas são da classe NP-difícil [Karp, 1972; Garey e Johnson, 1979]. A Figura 1 mostra um exemplo de empacotamento do 2BP, onde diversos itens, representados por diferentes cores, são alocados em 4 caixas.

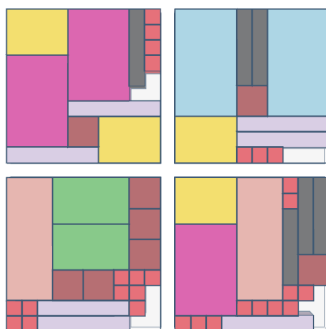


Figura 1: Exemplo de empacotamento bidimensional com 4 caixas.

Várias aplicações industriais têm interesse particular no 2BP. Dentre elas, podemos citar aplicações de carregamento de objetos sensíveis em caminhões e aviões, armazenamento de mercadorias, empacotamento de encomendas, reconfiguração dinâmica de hardware e carregamento de paletes. Por exemplo, quando um editor precisa preencher as páginas de um jornal, ele tem que considerar como rearranjar e posicionar os retângulos de textos e propagandas, ou seja, empacotamento de itens em que uma de suas faces deve permanecer obrigatoriamente para cima. Apesar de ser uma simplificação do mundo real, as heurísticas que encontram boas soluções para os problemas de empacotamento na versão clássica são aplicadas com sucesso em versões com restrições adicionais que modelam situações reais [Parreño et al., 2010b; Kang et al., 2012; Li et al., 2014; Saraiva et al., 2015; Boyar et al., 2016; Moura e Bortfeldt, 2017]. Além disso, o problema aparece como subparte de outros problemas mais complexos como os de agendamento, corte e particionamento.

Apesar do avanço da computação e da difusão de ambientes heterogêneos, os métodos exatos para os problemas de empacotamento, tal como [Martello e Vigo, 1998], são inviáveis de serem executados com instâncias grandes. Portanto, uma das alternativas práticas para a solução deste tipo de problema é a utilização de heurísticas e meta-heurísticas. Dessa forma, [Lodi et al., 2002a] desenvolveram um algoritmo Busca Tabu baseado em procedimentos construtivos para o 2BP. [Faroe et al., 2003] propuseram uma Busca Local Guiada para o 2BP com orientação fixa, tal que o algoritmo se baseia em soluções iterativas do problema de satisfação de restrições. [Boschetti e Mingozzi, 2003a,b] propuseram uma heurística construtiva que determina uma pontuação para cada item, considerando o empacotamento dos itens de acordo com sua pontuação em ordem de-

crescente. Assim, cada iteração atualiza os valores dos itens de acordo com sua estratégia específica de ranqueamento até que o critério de parada seja atingido. [Monaci e Toth, 2006] apresentaram uma heurística baseada em cobertura de conjuntos, onde a primeira fase gera uma coluna de itens para o empacotamento, assumindo que não há rotação.

Abordagens mais recentes incluem a heurística híbrida GRASP/VND que foi proposta em [Parreño et al., 2010a]. A fase de construção utiliza um valor máximo para as caixas, onde nem todos os itens precisam ser empacotados. O algoritmo utiliza uma estrutura VND (Variable Neighborhood Descent) com diversas estratégias de reempacotamento para obter uma solução onde todos os itens são empacotados. O espaço vazio das caixas abertas é modelado através de uma estratégia com espaços maximais. O trabalho [Gonçalves e Resende, 2013] introduziu um algoritmo genético de chaves aleatórias viciadas para o 2BP considerando itens que podem ser rotacionados ou não. Os autores utilizaram um algoritmo guloso para efetuar o empacotamento dos itens, baseando-se na mesma estratégia de espaços máximas do trabalho citado anteriormente. [Óscar Oliveira e Gamboa, 2019] apresentaram uma heurística iterativa de múltiplos recomeços para o 2BP.

Neste contexto, este trabalho estende o algoritmo proposto em [Zudio et al., 2018] para o problema clássico de empacotamento tridimensional, apresentando novos resultados para o 2BP com itens que podem ser orientados ou não. A abordagem proposta, denominada BRKGA/VCD, é um algoritmo genético de chaves aleatórias viciadas (BRKGA) [Gonçalves e Resende, 2011] que utiliza um método inspirado no Variable Neighborhood Descent (VND) [Caporossi et al., 2016] para acelerar o processo evolutivo da meta-heurística. O novo método aplicado, chamado de Variable Cross Descent (VCD), objetiva gerar configurações de empacotamento de itens parcialmente ordenados com alta probabilidade. O VCD é aplicado durante a geração da população inicial e mutantes, a fim de refinar a qualidade das soluções geradas nessa fase. O BRKGA/VCD foi submetido a um extensivo teste empírico composto de 500 instâncias, que são a base de teste para diversos trabalhos da literatura. O resultado obtido pelo BRKGA/VCD foi comparado com a de outros algoritmos no estado da arte para o 2BP em termos de qualidade da solução obtida. Considera-se o caso em que os itens têm orientação fixa ou admitem a possibilidade de rotação. Os dados mostram que o BRKGA/VCD provê soluções equivalentes ou melhores que os métodos comparados.

2. Algoritmo BRKGA/VCD

O algoritmo genético de chaves aleatórias viciadas (BRKGA) foi introduzido por [Gonçalves e Resende, 2011] para problemas de otimização combinatória. Esta meta-heurística é uma variante do algoritmo genético de chaves aleatórias (RKGA) [Bean, 1994].

A ideia principal é evoluir uma população codificada. Os indivíduos são representados por cromossomos, denominado chaves aleatórias ou *random-keys*, na forma de vetores de números reais no intervalo $[0, 1]$. A implementação desse processo requer um componente chamado decodificador. Este componente deve receber um vetor de *random-keys* para fornecer uma representação do indivíduo que pode ser avaliada. O decodificador é o único componente que depende das características do problema. Assim, duas implementações da meta-heurística para diferentes problemas só necessitam de decodificadores diferentes.

Para o 2BP, o decodificador é composto de dois componentes: um método de ordenação e um algoritmo guloso para realizar o empacotamento dos itens. Tal algoritmo guloso constrói uma solução iterativamente, recebendo uma permutação de inteiros que representa a sequência em que os itens serão empacotados sequencialmente. O método de ordenação deve receber um vetor com n *random-keys*, um para cada item da instância, para fornecer uma permutação de itens. As chaves devem ser ordenadas de modo que o item a_i é empacotado antes do item a_j se e somente se $k_i < k_j$, $i, j \in [1, \dots, n]$, onde k_i e k_j são as respectivas *random-keys* associadas.

Após o processo de ordenação, a permutação associada ao cromossomo do indivíduo é passada como parâmetro de entrada para o algoritmo guloso, denominado Distance to the Front-Top-Right Corner (DFTRC). Sabendo que uma caixa aberta da solução corrente é uma que contém itens empacotados, para cada iteração k do DFTRC, procuramos em cada caixa aberta por áreas que podem conter o respectivo item a_k , $k \in [1, \dots, n]$. Se nenhuma caixa aberta até o momento pode conter o item, então uma nova caixa é introduzida na solução com a_k empacotado. No passo de empacotamento, áreas maximais são utilizadas para modelar o espaço vazio de cada caixa aberta. Uma área maximal é a maior área retangular possível dentro da caixa aberta que não há itens. Todas as áreas maximais que podem conter o item da iteração corrente são candidatas para a escolha da localização do empacotamento. O DFTRC seleciona a área maximal que maximiza a distância entre o canto superior direito frontal do item para o respectivo canto superior direito frontal da caixa. O DFTRC termina suas iterações quando todos os itens forem empacotados.

Caso o problema permita a rotação dos itens, então o DFTRC considera as duas possíveis orientações de cada item, selecionando o candidato que maximiza a distância entre o canto superior direito frontal do item para o respectivo canto da caixa na possível orientação que apresenta a maior distância. Para a seleção final da orientação do item, um vetor de n *random-keys* adicionais é usado, totalizando $2n$ chaves neste caso. Quando as duas orientações são possíveis durante o empacotamento na determinada área maximal, a escolha é baseada no valor da respectiva chave adicional de acordo com sua relação com o valor 0,5.

No BRKGA original, a geração da população inicial e mutantes se trata de gerar vetores completamente aleatórios de *random-keys*. No método proposto neste trabalho, a geração é feita com o algoritmo Variable Cross Descent (VCD). O VCD é um algoritmo inspirado no VND que é aplicado na geração da população inicial e mutantes, com o objetivo de melhorar a qualidade média das soluções iniciais e de acelerar o processo de evolução do BRKGA para os problemas de empacotamento. Um dos maiores desafios de implementar o BRKGA é sintetizar estratégias algorítmicas que sejam capazes de utilizar a codificação em forma de *random-keys* de forma vantajosa. O VCD é uma estratégia algorítmica que é capaz de utilizar a codificação em forma de *random-keys* em conjunto com o método de reprodução do BRKGA para sintetizar soluções que têm uma alta probabilidade de apresentar uma boa qualidade comparada ao método aleatório.

Geralmente, a qualidade média das soluções geradas através de sequências aleatórias é baixa. Através de testes empíricos, observa-se que as soluções de alta qualidade são geradas através de permutações de entrada que realizam o empacotamento de várias subsequências de itens ordenados pelo maior volume, altura ou largura. Quando ordenamos os itens diretamente por maior volume, largura ou altura para obter a respectiva permutação, a qualidade de cada solução construída pelo DFTRC com essas sequências é superior comparada com a média dos indivíduos aleatórios. Dessa forma, a ideia do VCD é gerar vetores de *random-keys* que codificam permutações que com várias subsequências de itens parcialmente ordenadas. Consequentemente, o processo de evolução do BRKGA deve acelerar.

Para aplicar o VCD, devemos executar um processo preliminar para introduzir novos indivíduos que codificam permutações de itens ordenados para o empacotamento. O primeiro passo é ordenar os itens de entrada pelo maior volume, altura e largura para obter 3 permutações que representam as respectivas sequências de empacotamento. Logo após, este processo deve utilizar o DFTRC para empacotar os itens com cada permutação gerada como entrada. Então, o procedimento deve codificar estas soluções em indivíduos chamados **Universais** através da equação $r_i = \frac{p_i}{n}$, $1 \leq i \leq n$, tal que r_i é a *random-key* do i -ésimo item e p_i é seu respectivo termo da permutação. Caso o problema admita rotação, o vetor adicional de *random-keys* deve ser gerado

aleatoriamente. Finalmente, os universais devem ser mantidos em uma variável auxiliar que os armazena ordenados do melhor para o pior durante todo o processo do BRKGA/VCD.

O VCD recebe como parâmetro de entrada uma população a ser preenchida, um tamanho alvo para esta população e os universais ordenados do melhor para o pior. Fora do processo iterativo da meta-heurística, o VCD é utilizado para gerar a população inicial. Durante as iterações da meta-heurística, o VCD é utilizado para gerar os mutantes. O processo de decodificação tem alto custo computacional comparado com os outros procedimentos do BRKGA. Geralmente, os universais têm qualidade superior quando comparados à média de um indivíduo aleatório. Então, durante o VCD sempre assumimos que cada universal é o respectivo parente elite durante a reprodução, tal que o parente elite é aquele que tem maior qualidade. O primeiro passo de uma iteração do VCD é sintetizar um indivíduo completamente aleatório, que é introduzido na população. Em seguida, o VCD sintetiza novos indivíduos através da reprodução dos universais com indivíduos aleatórios com uma estratégia similar à troca de vizinhança do VND, sempre introduzindo o novo filho gerado na população.

O Algoritmo 1 detalha o VCD, tal que o método *Reproduzir* é o processo de reprodução do BRKGA. Dessa forma, o BRKGA/VCD é o BRKGA original utilizando o Algoritmo 1 para gerar a população inicial e mutantes.

Algorithm 1 Variable Cross Descent

```

função VCD(População:  $P, U$ ; Tamanho:  $q$ ; Taxa da reprodução:  $\rho$ ; Itens:  $I$ ; Dimensões:  $(W, H, D)$ )
1: enquanto  $|P| < q$  faça                                ▷ Enquanto a população não estiver preenchida
2:    $i \leftarrow$  Novo indivíduo completamente aleatório
3:    $s \leftarrow$  DECODIFICA( $i, I, (W, H, D)$ )
4:   Aptidão de  $i, A \leftarrow$  Avaliação de  $s$ 
5:    $P \leftarrow P \cup i$                                        ▷ Introduz  $i$  na população já com avaliação
6:    $k \leftarrow 1$                                              ▷ Procedimento inspirado por VND
7:   enquanto  $k \leq |U|$  e tamanho de  $P < q$  faça
8:      $f \leftarrow$  REPRODUZIR( $U_k, i, \rho$ )                    ▷  $U$  armazena os Universais
9:      $s \leftarrow$  DECODIFICA( $f, I, (W, H, D)$ )                ▷ Filho do universal  $k$  com  $i$ 
10:    Aptidão de  $f \leftarrow$  Avaliação de  $s$ 
11:     $P \leftarrow P \cup f$                                      ▷ Introduz o filho em  $P$  já com avaliação
12:    se Aptidão de  $f < A$  então
13:       $A \leftarrow$  Aptidão de  $f$ 
14:       $k \leftarrow 0$ 
15:    senão  $k \leftarrow k + 1$ 
16: retorna  $P$ 

```

3. Resultados Computacionais

A base de testes utilizada neste trabalho é composta por 500 instâncias do 2BP detalhadas em [Martello e Vigo, 1998]. As instâncias são organizadas em 10 classes com 10 instâncias para cada quantidade de itens $n \in \{20, 40, 60, 80, 100\}$. Esta base se trata de uma extensão das instâncias propostas em [Berkey e Wang, 1987]. O BRKGA/VCD proposto foi implementado utilizando a linguagem de programação *C++11*, e foi compilado com o *GCC* do *GNU Compiler Collection*. O ambiente computacional utilizado em todos os testes neste trabalho consiste de um notebook munido da seguinte configuração: processador Intel Core i5-240CM @2.5 GHz, 8 GB de memória RAM e sistema operacional Ubuntu 16.01. Em cada experimento, cada instância foi executada 10 vezes com sementes sequenciais no intervalo $[1, 10]$ para o gerador de números pseudoaleatórios, a fim de

controlar o âmbito de execução. A Tabela 1 relaciona os parâmetros calibráveis do BRKGA/VCD com os respectivos valores. Tais parâmetros foram calibrados em um teste preliminar.

Tabela 1: Configuração do BRKGA/VCD.

Componente	Configuração
Critério de parada	200 gerações
Tamanho da população	$P_{max} = 30 \times n$
Fração TOP	$\epsilon = 0,1$
Fração BOT	$\omega = 0,15$
Taxa de elitismo	$\rho = 0,7$

Os resultados obtidos neste experimento foram comparados com os dados reportados nos trabalhos da literatura descritos na Tabela 2, que associa os algoritmos comparados. Estes trabalhos utilizam o mesmo conjunto de instâncias mencionado como base de teste para os experimentos computacionais. A tabela relaciona o respectivo pseudônimo do algoritmo com o trabalho de referência e a estratégia utilizada para resolver os problemas de empacotamento. Todos os dados reportados neste trabalho são os mesmos apresentados pelos autores das respectivas referências.

Tabela 2: Métodos literatura utilizados na comparação com o BRKGA/VCD.

Algoritmo	Referência	Estratégia
TS3	[Lodi et al., 2002b]	Busca Tabu
GLS	[Faroe et al., 2003]	Busca Local Guiada
GVND	[Parreño et al., 2010a]	GRASP/VND híbrido
*BRKGA	[Gonçalves e Resende, 2013]	<i>Biased random-key Genetic Algorithm</i>
HPB	[Boschetti e Mingozzi, 2003b]	Heurística
SCH	[Monaci e Toth, 2006]	Heurística baseada em cobertura de conjunto
ASH	[Óscar Oliveira e Gamboa, 2019]	Heurística Iterativa

*Único trabalho que reporta soluções com a possibilidade de rotação.

O BRKGA/VCD foi comparado com o BRKGA diretamente. Ambos os algoritmos foram executados para as mesmas instâncias com o mesmo critério de parada, mas testados em diferentes ambientes computacionais. A Tabela 3 mostra o acumulado de caixas obtidas em todas as instâncias do 2BP sem rotação, através das gerações do BRKGA/VCD. A tabela também mostra a mesma informação para o BRKGA original com os valores reportados no respectivo trabalho citado.

Tabela 3: Acumulado de caixas obtidas em todas as instâncias do 2BP sem rotação através das gerações.

Método	Número de gerações									
	1	10	25	50	75	100	125	150	175	200
BRKGA/VCD	7286	7280	7251	7242	7232	7231	7231	7230	7230	7230
BRKGA	7359	7299	7261	7248	7245	7241	7234	7234	7234	7234

Considerando que o número de avaliações do BRKGA/VCD e do BRKGA são o mesmo,

o método proposto neste trabalho foi capaz de obter um acumulado final de caixas melhor com 75 iterações comparado com 200 iterações do BRKGA. Podemos observar também que o acúmulo de caixas utilizadas pela melhor solução na população inicial é significativamente menor que o reportado pelo método original. A quantidade de caixas economizadas através das gerações mostra que o VCD acelera consideravelmente o processo evolutivo do BRKGA. Além disso, as soluções encontradas pelo BRKGA/VCD são melhores que o BRKGA em qualquer geração para este conjunto de instâncias. Além disso, o tempo de execução para gerar as soluções constitui 99,9% do tempo total de execução. Assim, o VCD é uma estratégia algorítmica que impõe um *overhead* insignificante comparado com a execução do método original.

As Tabelas 4 e 5 apresentam os resultados do BRKGA/VCD para as instâncias do 2BP sem rotação e com rotação respectivamente, porém detalhando a média de cada subconjunto composto por 10 instâncias. As colunas 1 e 2 relacionam a classe *Cl* e a quantidade de itens de cada subconjunto, notando-se que os elementos da tabela são as médias. A coluna 3 reporta a média 200g do tempo total de execução em CPU do BRKGA/VCD para atingir o critério de parada de 200 gerações e a coluna 4 exibe a solução média *Z* obtida. As demais colunas reportam os resultados dos trabalhos da literatura. A última linha da tabela apresenta o acumulado de caixas obtidas em todas as instâncias.

Os resultados para o 2BP sem rotação mostram que o BRKGA/VCD provê soluções melhores ou equivalentes a todas as reportadas pelos trabalhos comparados da literatura. Todos os prováveis ótimos globais são encontrados para este conjunto em pouco tempo de execução de CPU no ambiente computacional utilizado. Além disso, vários resultados obtidos melhoram a qualidade reportada pelos trabalhos da literatura. O acumulado final de caixas mostra que o BRKGA/VCD obtém soluções melhores que todos os outros métodos comparados para todas as instâncias deste conjunto. O teste de Wilcoxon pareado mostra que as diferenças observadas na tabela implicam que o BRKGA-VCD é significativamente melhor que todos métodos comparados, com exceção do BRKGA. Para este último algoritmo, o teste diz que os resultados da tabela não provê uma diferença estatisticamente significativa. Porém, a Tabela 3 mostra que o BRKGA/VCD é capaz de obter um acumulado final melhor que o BRKGA com 125 iterações comparado com 200 do método original.

Para os resultados com rotação, o BRKGA/VCD economizou somente 1 caixa nas 500 instâncias bidimensionais testadas com rotação. Assim, os dois métodos são equivalentes para este conjunto de instâncias. Porém, a Tabela 3 sugere que o BRKGA/VCD é capaz de obter estas soluções em uma quantidade inferior de iterações comparado ao BRKGA nest conjunto de instâncias. Então, há uma alta probabilidade dos valores reportados serem um ótimo local de alta qualidade para itens com rotação.

Vale observar que o BRKGA/VCD e o BRKGA efetuam a mesma quantidade de avaliações por iteração. Isto se deve ao algoritmo VCD que decodifica o mesmo número de soluções que o método original, pois os parâmetros calibráveis do BRKGA/VCD receberam os mesmos valores utilizados em Gonçalves e Resende [2013]. Além disso, o tempo de execução para decodificar as soluções constitui 99,9% do tempo total de execução. Assim, o VCD é uma estratégia algorítmica que impõe um *overhead* insignificante comparado com a execução do método original. Quanto ao tempo de execução em CPU, este trabalho não apresenta nenhuma análise ou observação comparativa entre o algoritmo proposto e os métodos comparados da literatura, pois todos os algoritmos foram executados em ambientes computacionais distintos.

4. Conclusão

Este trabalho aplicou a heurística BRKGA/VCD para o problema clássico de empacotamento bidimensional, considerando que os itens podem ser rotacionados ou não. O algoritmo é

Tabela 4: Resultados do BRKGA/VCD e dos algoritmos comparados para o 2BP sem rotação.

Cl	n	200g(s)	Z	TS3	GLS	HBP	SCH	GVND	BRKGA	ASH
1	20	2,5	7,1	7,1	7,1	7,1	7,1	7,1	7,1	7,1
	40	10,0	13,4	13,5	13,4	13,4	13,4	13,4	13,4	13,4
	60	23,3	20,0	20,1	20,1	20,1	20,0	20,0	20,0	20,0
	80	40,2	27,5	28,2	27,5	27,5	27,5	27,5	27,5	27,5
	100	64,4	31,7	32,6	32,1	31,8	31,7	31,7	31,7	31,7
2	20	4,7	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0
	40	15,9	1,9	2,0	1,9	1,9	1,9	1,9	1,9	1,9
	60	38,7	2,5	2,7	2,5	2,5	2,5	2,5	2,5	2,5
	80	69,0	3,1	3,3	3,1	3,1	3,1	3,1	3,1	3,2
	100	107,0	3,9	4,0	3,9	3,9	3,9	3,9	3,9	3,9
3	20	3,2	5,1	5,5	5,1	5,1	5,1	5,1	5,1	5,1
	40	12,8	9,4	9,7	9,4	9,5	9,4	9,4	9,4	9,4
	60	29,4	13,9	14,0	14,0	14,0	13,9	13,9	13,9	14,0
	80	58,6	18,9	19,8	19,1	19,1	18,9	18,9	18,9	19,2
	100	120,9	22,3	23,6	22,6	22,6	22,3	22,3	22,3	22,5
4	20	7,4	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0
	40	28,2	1,9	1,9	1,9	1,9	1,9	1,9	1,9	1,9
	60	56,7	2,4	2,6	2,5	2,5	2,5	2,5	2,5	2,5
	80	100,0	3,1	3,3	3,3	3,3	3,2	3,1	3,1	3,2
	100	167,0	3,7	4,0	3,8	3,8	3,8	3,8	3,7	3,8
5	20	3,2	6,5	6,6	6,5	6,5	6,5	6,5	6,5	6,5
	40	13,1	11,9	11,9	11,9	11,9	11,9	11,9	11,9	11,9
	60	30,3	18,0	18,2	18,1	18,0	18,0	18,0	18,0	18,1
	80	54,9	24,7	25,1	24,9	24,8	24,7	24,7	24,7	24,7
	100	88,7	28,1	29,5	28,8	28,7	28,2	28,2	28,1	28,7
6	20	6,3	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0
	40	28,1	1,6	1,9	1,8	1,8	1,7	1,7	1,6	1,8
	60	69,2	2,1	2,2	2,2	2,1	2,1	2,1	2,1	2,2
	80	117,1	3,0	3,0	3,0	3,0	3,0	3,0	3,0	3,0
	100	220,0	3,2	3,4	3,4	3,4	3,4	3,4	3,3	3,4
7	20	2,7	5,5	5,5	5,5	5,5	5,5	5,5	5,5	5,5
	40	11,9	11,1	11,4	11,3	11,1	11,1	11,1	11,1	11,2
	60	28,0	15,8	16,2	15,9	16,0	15,8	15,9	15,8	15,9
	80	53,4	23,1	23,2	23,2	23,2	23,2	23,2	23,2	23,2
	100	85,0	27,1	27,7	27,5	27,4	27,1	27,1	27,1	27,3
8	20	2,4	5,8	5,8	5,8	5,8	5,8	5,8	5,8	5,8
	40	12,0	11,3	11,4	11,4	11,3	11,3	11,3	11,3	11,3
	60	28,2	16,1	16,2	16,3	16,2	16,2	16,1	16,1	16,2
	80	53,3	22,4	22,6	22,5	22,6	22,4	22,4	22,4	22,6
	100	84,5	27,7	28,4	28,1	28,0	27,9	27,8	27,8	27,8
9	20	2,6	14,3	14,3	14,3	14,3	14,3	14,3	14,3	14,3
	40	11,8	27,8	27,8	27,8	27,8	27,8	27,8	27,8	27,8
	60	29,1	43,7	43,8	43,7	43,7	43,7	43,7	43,7	43,7
	80	51,8	57,7	57,7	57,7	57,7	57,7	57,7	57,7	57,7
	100	90,4	69,5	69,5	69,5	69,5	69,5	69,5	69,5	69,5
10	20	3,3	4,2	4,3	4,2	4,3	4,2	4,2	4,2	4,3
	40	13,4	7,4	7,5	7,4	7,4	7,4	7,4	7,4	7,4
	60	30,9	10,0	10,4	10,2	10,2	10,1	10,0	10,0	10,2
	80	60,2	12,8	13,0	13,0	13,0	12,8	12,9	12,8	13,0
	100	97,3	15,8	16,6	16,2	16,2	15,9	15,9	15,8	16,1
(1 – 10)			7230	7360	7284	7275	7243	7241	7234	7269

*Valores em negrito correspondem às melhores soluções de cada conjunto de instâncias

Tabela 5: Resultados do BRKGA-VCD e do algoritmo comparado para o 2BP com rotação.

CI	n	200g(s)	Z	BRKGA
1	20	0,2	6,6	6,6
	40	0,8	12,8	12,8
	60	2,1	19,5	19,5
	80	3,7	27,0	27,0
	100	5,9	31,3	31,3
2	20	0,7	1,0	1,0
	40	2,2	1,9	1,9
	60	4,8	2,5	2,5
	80	8,5	3,1	3,1
	100	12,8	3,9	3,9
3	20	0,3	4,7	4,7
	40	1,1	9,2	9,2
	60	2,6	13,4	13,4
	80	5,3	18,2	18,2
	100	8,8	21,9	22,0
4	20	0,9	1,0	1,0
	40	2,9	1,9	1,9
	60	7,2	2,3	2,3
	80	11,7	3,1	3,1
	100	18,1	3,7	3,7
5	20	0,3	5,9	5,9
	40	1,2	11,4	11,4
	60	3,0	17,2	17,2
	80	5,6	23,9	23,9
	100	8,5	27,7	27,7
6	20	0,9	1,0	1,0
	40	3,4	1,6	1,6
	60	8,1	2,1	2,1
	80	14,3	3,0	3,0
	100	27,3	3,2	3,2
7	20	0,3	5,2	5,2
	40	1,2	10,2	10,2
	60	2,6	14,6	14,6
	80	5,2	20,8	20,8
	100	8,0	25,0	25,0
8	20	0,3	5,3	5,3
	40	1,1	10,3	10,3
	60	2,5	14,7	14,7
	80	5,3	20,4	20,4
	100	9,1	25,2	25,2
9	20	0,3	14,3	14,3
	40	1,3	27,5	27,5
	60	3,2	43,5	43,5
	80	5,8	57,3	57,3
	100	10,1	69,3	69,3
10	20	0,4	4,1	4,1
	40	1,5	7,2	7,2
	60	3,4	9,9	9,9
	80	6,4	12,5	12,5
	100	10,1	15,4	15,4
(1 – 10)			6987	6988

*O valor em negrito corresponde ao único resultado melhorado.

baseado na meta-heurística BRKGA, e utiliza um algoritmo guloso para empacotar os itens. Através de um conjunto de 500 instâncias, as quais compõem a base de teste padrão para vários trabalhos encontrados na literatura, o BRKGA/VCD foi comparado com outras abordagens encontradas na literatura. Os experimentos mostraram que o BRKGA proposto aliado com a abordagem VCD obtém resultados equivalentes ou melhores aos reportados na literatura. Finalmente, os dados também mostram que o BRKGA-VCD é capaz de obter resultados de alta qualidade para instâncias de larga escala em pouco tempo de execução em CPU, quando executados em um notebook padrão munido com um processador *single core* de 2,5GHz.

Como trabalhos futuros, pretende-se estender a base de testes padrão com instâncias que utilizam uma quantidade maior de itens. Objetiva-se estudar outras variantes do 2BP, por exemplo, considerar itens com peso ou caixas não homogêneas. Além disso, pretende-se estender as abordagens propostas para a execução em ambientes heterogêneos CPU-GPU.

5. Agradecimentos

Este trabalho foi concebido utilizando os recursos computacionais do Laboratório de Inteligência Computacional (LABIC) do Instituto de Computação (IC) da Universidade Federal Fluminense (UFF). O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Referências

- Bean, J. C. (1994). Genetic algorithms and random keys for sequencing and optimization. *ORSA Journal on computing*, 6(2):154–160.
- Berkey, J. O. e Wang, P. Y. (1987). Two-dimensional finite bin-packing algorithms. *Journal of the operational research society*, 38(5):423–429.
- Boschetti, M. A. e Mingozzi, A. (2003a). The two-dimensional finite bin packing problem. part i: New lower bounds for the oriented case. *Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, 1(1):27–42.
- Boschetti, M. A. e Mingozzi, A. (2003b). The two-dimensional finite bin packing problem. part ii: New lower and upper bounds. *Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, 1(2):135–147.
- Boyar, J., Shahin, K., Larsen, K. S., e López-Ortiz, A. (2016). Online bin packing with advice. *Algorithmica*, 74(1):507–527.
- Caporossi, G., Hansen, P., e Mladenović, N. (2016). Variable neighborhood search. In Siarry, P., editor, *Metaheuristics*, p. 77–98. Springer, Cham, Switzerland, 1 edition.
- Faroe, O., Pisinger, D., e Zachariasen, M. (2003). Guided local search for the three-dimensional bin-packing problem. *Journal on computing*, 15(3):267–283.
- Garey, M. R. e Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, 1 edition. ISBN 0716710447.
- Gonçalves, J. F. e Resende, M. G. (2011). Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics*, 17(5):487–525.
- Gonçalves, J. F. e Resende, M. G. (2013). A biased random key genetic algorithm for 2d and 3d bin packing problems. *International Journal of Production Economics*, 145(2):500–510.

- Kang, K., Moon, I., e Wang, H. (2012). A hybrid genetic algorithm with a new packing strategy for the three-dimensional bin packing problem. *Applied Mathematics and Computation*, 219(3): 1287–1299.
- Karp, R. M. (1972). Reducibility among combinatorial problems. In Miller, R. E., Thatcher, J. W., e Bohlinger, J. D., editors, *Complexity of computer computations*, p. 85–103. Springer, New York, 1 edition.
- Li, X., Zhao, Z., e Zhang, K. (2014). A genetic algorithm for the three-dimensional bin packing problem with heterogeneous bins. In *Proc. of the 2014 Indus. and Sys. Eng. Research Conference*, p. 2039–2048, Peachtree Corners. Industrial and Systems Engineering Research Conference, 2014, Montréal, IISE.
- Lodi, A., Martello, S., e Monaci, M. (2002a). Two-dimensional packing problems: A survey. *European journal of operational research*, 141(2):241–252.
- Lodi, A., Martello, S., e Vigo, D. (2002b). Heuristic algorithms for the three-dimensional bin packing problem. *European Journal of Operational Research*, 141(2):410–420.
- Martello, S. e Vigo, D. (1998). Exact solution of the two-dimensional finite bin packing problem. *Management science*, 44(3):388–399.
- Monaci, M. e Toth, P. (2006). A set-covering-based heuristic approach for bin-packing problems. *Journal on Computing*, 18(1):71–85.
- Moura, A. e Bortfeldt, A. (2017). A two-stage packing problem procedure. *International Transactions in Operational Research*, 24(1-2):43–58.
- Parreño, F., Alvarez-Valdés, R., Oliveira, J., e Tamarit, J. M. (2010a). A hybrid grasp/vnd algorithm for two-and three-dimensional bin packing. *Annals of Operations Research*, 179(1):203–220.
- Parreño, F., Alvarez-Valdés, R., Oliveira, J., e Tamarit, J. M. (2010b). Neighborhood structures for the container loading problem: a vns implementation. *Journal of Heuristics*, 16(1):1–22.
- Saraiva, R. D., Nepomuceno, N., e Pinheiro, P. R. (2015). A layer-building algorithm for the three-dimensional multiple bin packing problem: a case study in an automotive company. *IFAC-PapersOnLine*, 48(3):490–495.
- Wäscher, G., Haußner, H., e Schumann, H. (2007). An improved typology of cutting and packing problems. *European journal of operational research*, 183(3):1109–1130.
- Zudio, A., Costa, D., Masquio, B., Coelho, I., e Pinto, P. (2018). Brkga/vnd hybrid algorithm for the classic three-dimensional bin packing problem. *Electronic Notes in Discrete Mathematics*, 66:175–182.
- Óscar Oliveira e Gamboa, D. (2019). Adaptive sequence-based heuristic for the two-dimensional non-guillotine bin packing problem. In *Proceedings of HIS 2018*, p. 370–375, Porto, Portugal. International Conference on Hybrid Intelligent Systems, 18th, 2018, Porto, Springer.