LIII Simpósio Brasileiro de Pesquisa Operacional
João Pessoa - PB, 3 a 5 de novembro de 2021

Aprendizado por reforço aplicado ao problema de empacotamento de peças irregulares em faixas

Petra Maria Bartmeyer, Larissa Tebaldi Oliveira, Franklina Maria Bragion Toledo
Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo (USP)
Av. Trabalhador São-carlense, 13566-590, São Carlos - SP, Brasil
(petra.bartmeyer,ltdo,fran)@icmc.usp.br

Aline Aparecida Souza Leão
Departamento de Matemática - Universidade Estadual de Londrina
Rod. Celso Garcia Cid, PR-445, km 380, 86057-970, Londrina-PR, Brasil
aasleao@uel.br

RESUMO

Este trabalho estuda os benefícios de estratégias de aprendizado de máquina para tratar o problema de empacotamento de peças irregulares em faixa. Em específico, é estudado o benefício da transferência de aprendizagem aplicada à estratégia de aprendizado por reforço. A escolha do método de transferência de aprendizado se deve ao número de exemplares que compartilham peças semelhantes, o que gera a hipótese de que o aprendizado de um exemplar poderia ser utilizado como ponto de partida para o aprendizado de outros. Testes estatísticos mostram que a transferência de aprendizado permite atingir desempenho semelhante ao método de aprendizado por reforço puro com um sexto do tempo de execução. Estudos computacionais mostram os benefícios trazidos pela introdução da transferência de aprendizado, em especial, para exemplares com peças côncavas.

PALAVRAS CHAVE. Empacotamento de peças irregulares em faixas. Aprendizado por reforço. Transferência de aprendizado.

IC - Inteligência Computacional, OA - Outras Aplicações em PO

ABSTRACT

This research studies the benefits of addressing the irregular strip packing problem with machine learning strategies. In particular, the benefit of combining transfer learning and reinforcement learning method. The choice of the transfer learning strategy is due to the number of instances that share similar pieces, which generates the hypothesis that the learning of one instance could be used as a starting point for the reinforcement learning method in other cases. Statistical tests show that the transfer learning strategy allows achieving similar performance to the pure reinforcement learning method with one-sixth of the execution time. Also, computational studies illustrate the benefits of introducing transfer learning strategies, especially for concave pieces.

KEYWORDS. Irregular strip packing problem. Reinforcement learning. Transfer learning.

CI - Computational Intelligence, OA - O ther OR applications

1. Introdução

Problemas de corte e empacotamento de peças irregulares consistem em empacotar um conjunto de peças regulares e irregulares em objetos maiores. De acordo com a definição de Bennell e Oliveira [2009], uma peça é caracterizada como irregular se pelo menos três parâmetros são necessários para identificá-la. Os objetos maiores podem ser circulares, retangulares de dimensão fixa, ou de uma ou mais dimensões variáveis, bem como irregulares. Neste trabalho, é considerado o problema de empacotamento de peças irregulares bidimensionais em faixas, também conhecido como *nesting*, em que as peças são empacotadas em um objeto retangular de altura fixa e comprimento variável. De acordo com a tipologia proposta em Wäscher et al. [2007], o problema de empacotamento em faixas é classificado como um problema bidimensional com uma dimensão aberta.

Estes problemas têm diversas aplicações industriais como na fabricação de móveis, de roupas, de artefatos de vidro e em metalúrgicas. Devido à sua aplicabilidade industrial e dificuldade de resolução, diversos métodos heurísticos foram investigados na literatura. Uma revisão das estratégias heurísticas clássicas utilizadas para resolver problemas de empacotamento de peças irregulares é apresentada em Bennell e Oliveira [2009]. De modo geral, as heurísticas podem ser divididas em construtivas e de melhoria. Para as heurísticas construtivas são analisadas regras para o posicionamento das peças. O método mais utilizado nessas heurísticas é o *bottom-left*, em que as peças são ordenadas de acordo com algum critério, para então, serem alocadas uma a uma no objeto, sempre na posição mais a esquerda e mais abaixo possível. Um estudo sobre a sequência das peças é apresentado em Gomes e Oliveira [2002]. As heurísticas de melhoria envolvem mudanças na posição e orientação das peças, como os métodos de compactação e separação, sendo o último utilizado para remover infactibilidades [Bennell e Dowsland, 2001; Gomes e Oliveira, 2006]. Métodos que combinam diferentes estratégias com heurísticas construtivas e de melhoria também têm sido propostos, como meta-heurísticas [Elkeran, 2013] e modelos de programação inteira [Sato et al., 2019].

Dada a variabilidade de desempenho das meta-heurísticas para a resolução do problema de empacotamento de itens em faixas, Rakotonirainy [2020] propõe um método de aprendizado de máquina para selecionar a melhor heurística para cada exemplar, tomando como base as características dos mesmos. De fato, assim como ideias de aprendizado de máquina vem ganhando espaço na resolução de problemas de otimização combinatória [Bengio et al., 2020], a sua aplicabilidade à problemas de corte em empacotamento vem crescendo. Alguns exemplos são: o método baseado em aprendizado por reforço para o problema de empacotamento 3D de itens regulares em *bins* proposto por Hu et al. [2017] e o uso de métodos de aprendizado de máquina para prever a factibilidade do conjunto de itens a serem produzidos em cada lote [Gahm et al., 2021]. Contudo, no melhor do nosso conhecimento, ainda não existem métodos de aprendizado de máquina diretamente aplicados à alocação de peças em problemas de *nesting*. Nesse sentido, esse trabalho propõe duas abordagens baseadas em aprendizado de máquina para o problema de empacotamento de peças irregulares em faixas. A primeira utiliza puramente aprendizado por reforço, enquanto a segunda também explora o benefício de técnicas de transferência de aprendizado. Vale destacar, que a transferência de aprendizado é extremamente útil quando consideramos problemas que são resolvidos frequentemente com pequenas diferenças, por exemplo, no contexto de fabricação de roupas, em que os modelos e as quantidades mudam, mas muitas peças continuam as mesmas.

A organização deste trabalho se dá da seguinte forma: o problema de empacotamento de peças irregulares em faixas é definido na Seção 2, as estratégias de resolução são apresentadas na Seção 3, a Seção 4 contém os estudos computacionais comparando as estratégias de resolução,

seguida das conclusões da pesquisa apresentadas na Seção 5.

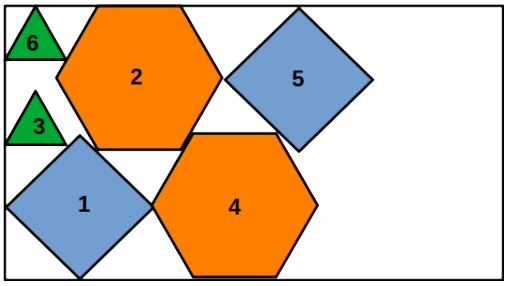
2. Problema de empacotamento de peças irregulares em faixas

O problema de empacotamento de peças irregulares em faixas consiste em dispor peças formadas por polígonos côncavos ou convexos em um objeto de altura fixa e comprimento variável. Uma solução factível para o problema deve garantir que não haja sobreposição entre as peças. Os requisitos de factibilidade para o problema ainda podem ser combinados com algum objetivo específico, como a minimização do comprimento utilizado para fazer a alocação, que é o objetivo considerado nesse trabalho.

Uma solução para o problema de *nesting* pode ser representada por uma sequência de peças a serem alocadas. A posição dessas peças no espaço bidimensional é dada conforme a regra de alocação escolhida. Exemplos de regras são *bottom-left* e *bottom-up* [Baker et al., 1980]. Para a pesquisa apresentada nesse trabalho, a regra de alocação *bottom-left* é utilizada.

Para a regra *bottom-left*, cada peça da sequência é alocada na posição factível mais a esquerda e mais abaixo no objeto, respeitando a posição das peças previamente alocadas. Na Figura 1(b), é ilustrada uma solução gerada pela regra *bottom-left* a partir da sequência definida na Figura 1(a), onde o número dentro de cada polígono representa a posição de cada peça na sequência de alocação.

(a) Sequência para alocação.



(b) Alocação gerada pela regra *bottom-left*.

Figura 1: Representação de uma sequência de alocação seguindo a regra *bottom-left*.

3. Estratégias de resolução

Neste trabalho, são analisadas duas estratégias de aprendizado de máquina fundamentadas nas ideias de aprendizado por reforço. Os métodos de aprendizado por reforço são baseados na análise da qualidade das decisões sequenciais realizadas. Nesse tipo de método, cada decisão tomada é avaliada e recebe recompensas ou penalidades a depender da qualidade da ação, sendo o objetivo maximizar a recompensa total. Para o problema estudado, a decisão a ser tomada é o tipo de peça alocada a cada posição da sequência utilizada pela regra de alocação *bottom-left*.

A técnica de aprendizado por reforço utilizada é denominada *Q-learning* [Watkins e Dayan, 1992] e seu desenvolvimento tem vários paralelos com as técnicas de programação dinâmica aproximada [Powell, 2016]. Em especial, a função de Bellman, utilizada para computar o valor da tomada de decisão a (qual peça alocar) no estado s (posição na sequência), é utilizada para computar

a recompensa de cada ação tomada. As funções de aprendizado com reforço podem ser desenhadas de várias formas, podendo ser diferenciadas pela qualidade da resposta. Por exemplo, uma função de recompensa no estilo +1/-1 penaliza e bonifica da mesma forma todas as soluções acima (ou abaixo) de um limitante, independente de quão boa (ou ruim) for a solução. Ou seja, usando recompensas do estilo +1/-1 para um exemplar onde o limitante superior é 27, duas soluções, de comprimento 25 e 23, respectivamente, teriam a mesma recompensa. Da mesma forma que soluções com comprimento 29 e 30 teriam a mesma penalização. Esse tipo de recompensa padrão pode não favorecer o surgimento de soluções cada vez mais próximas do valor ótimo. Por outro lado, quando consideramos recompensas relacionadas ao comprimento da solução, a função de recompensa pode necessitar de parâmetros que introduzam informações sobre o exemplar para computar a recompensa, como um limitante superior.

As recompensas podem ser distribuídas ao final de cada tomada de decisão, avaliando a qualidade a cada passo da construção da solução, ou ainda, ao final de todos os passos, quando a solução final é definida. Cada uma das formas de recompensa tem seus prós e contras. Recompensas a cada tomada de decisão permitem analisar decisões individualmente, contudo podem ser míopes sobre a relação da decisão para a qualidade da solução final. Por outro lado, recompensas baseadas na solução final tendem a dar a mesma contribuição para todas as decisões tomadas durante o processo, tornando mais difícil avaliar a contribuição de cada decisão para a solução final. Em ambos os casos, é esperado que o alto número de repetições do método supere as limitações da função de recompensa escolhida.

3.1. Matriz de aprendizado

Para o método *Q-learning*, o aprendizado é representado por uma matriz (Q) que guarda as recompensas de cada decisão tomada em cada estado. A matriz Q é retangular ($n \times m$) onde n é a quantidade de tipos de peças a serem alocados e m é a soma das demandas dos tipos de peças, ou seja, o tamanho total da sequência. Dessa forma, uma entrada Q_{ij} representa o benefício do uso de uma peça do tipo i na posição j da sequência de alocação. A cada solução gerada pelo método *Q-learning*, a matriz de aprendizado é atualizada de acordo com a qualidade da solução. A qualidade de uma solução é medida pelo comprimento da solução (CW), no limitante do comprimento do objeto original (OW) e no comprimento da melhor solução incumbente (BW), dado por:

$$Q_{ij} = Q_{ij} + \alpha; \qquad \text{se } (BW - CW) \geq 0 \text{ ou } (OW - CW) \geq 0, \tag{1}$$

$$Q_{ij} = Q_{ij} - \beta; \qquad \text{caso contrário.} \tag{2}$$

Uma nova solução é gerada com base nos valores de Q. Mais especificamente, cada elemento da sequência é escolhido por uma seleção do tipo roleta ponderada, em que o peso de cada escolha é baseado na sua possível contribuição para a solução, dado por:

$$p_i = e^{Q_{ij}}, \text{ se } i \text{ é factível.} \tag{3}$$

É importante ressaltar que somente as decisões factíveis são computadas, ou seja, somente os tipos de peças para os quais as demandas ainda não foram atendidas podem ser adicionados a sequência.

O método *Q-learning* utilizado está detalhado no Algoritmo 1, em que são dados de entrada: um limitante para o comprimento do objeto (OW), o comprimento da melhor solução incumbente (BW), a demanda de cada tipo de peça (vetor d), o número total de peças (m) e os valores de α e β utilizados nas Equações (1) e (2). A aleatoriedade do método fica por conta da seleção por roleta, enquanto a influência da matriz de aprendizado na decisão a ser tomada é dada pela ponderação de cada entrada da roleta.

Algoritmo 1 Aprendizado por reforço.

Dados de Entrada: OW, BW, d, m, α e β
procedimento ITERAÇÕES DE APRENDIZADO POR REFORÇO
    $Q \leftarrow \mathbf{0}$
    enquanto critério de parada não atingido **faça**
        $S \leftarrow \emptyset$
        para cada posição da solução (j)
            $p_i \leftarrow e^{Q_{ij}},\ \forall i : d_i \geq 0$ ▷ Calcular contribuição de cada peça factível
            $S_j \leftarrow i$ ▷ Seleção por roleta ponderada
            $d_i \leftarrow d_i - 1$ ▷ Atualiza a demanda da peça selecionada
        $CW \leftarrow BL(S)$ ▷ Calcula o comprimento da solução usando a regra *bottom-left*
        se $(BW - CW) \geq 0$ ou $(OW - CW) \geq 0$ **então** ▷ Atualização da matriz Q
            para cada entrada da sequência
                $Q_{ij} \leftarrow Q_{ij} + \alpha$
        senão
            para cada entrada da sequência
                $Q_{ij} \leftarrow Q_{ij} - \beta$
        se $CW \leq BW$ **então** ▷ Atualização da melhor solução incumbente
            $BW \leftarrow CW$

3.2. Uso de transferência de aprendizado

Observado que os exemplares do problema de *nesting* compartilham muitas peças similares, a ideia é criar uma matriz de aprendizado que possa ser passada de um exemplar para outro. A expectativa é que ao iniciar o algoritmo de aprendizado por reforço com uma matriz Q próxima à ideal, um menor número de iterações será necessário para a convergência da matriz de aprendizado. Em especial, estudamos o caso em que a transferência de aprendizado ocorre em um exemplar onde as peças são representações do contorno convexo (*rco*) de peças de outros exemplares (*blazewicz*). A diferença entre exemplares com peças convexas e não-convexas se mostra no esforço computacional demandado pela regra *bottom-left*. Experimentos computacionais preliminares, mostraram que enquanto exemplares com peças apenas convexas têm sua análise de alocação realizada em décimos de segundos, exemplares com peças não-convexas podem demandar dezenas de segundos.

Como a matriz Q é composta pelos tipos de peças (n) e a demanda total do exemplar (m), é necessário que a transferência de aprendizado seja realizada utilizando exemplares com as mesmas quantidades de peças, ou ainda, que a matriz Q seja redimensionada de acordo com as necessidades do novo exemplar.

As ideias implementadas para o método com transferência de aprendizado estão sumarizadas no Algoritmo 2. Note que a matriz Q gerada como saída do Algoritmo 1 é um parâmetro de entrada nesse método. Além disso, para que a matriz de aprendizado se adapte ao novo exemplar e o método supere possíveis *overfitting* devidos à matriz herdada, iterações puramente aleatórias são feitas a cada iteração do algoritmo. Os passos gerais do método proposto estão descritos na Figura 2.

4. Estudos computacionais

Os estudos computacionais foram realizados utilizando 10 exemplares da literatura, sendo os cinco exemplares *blazewicz(k)* $(k = 1, ..., 5)$ baseados nos exemplares de Błażewicz et al. [1993],

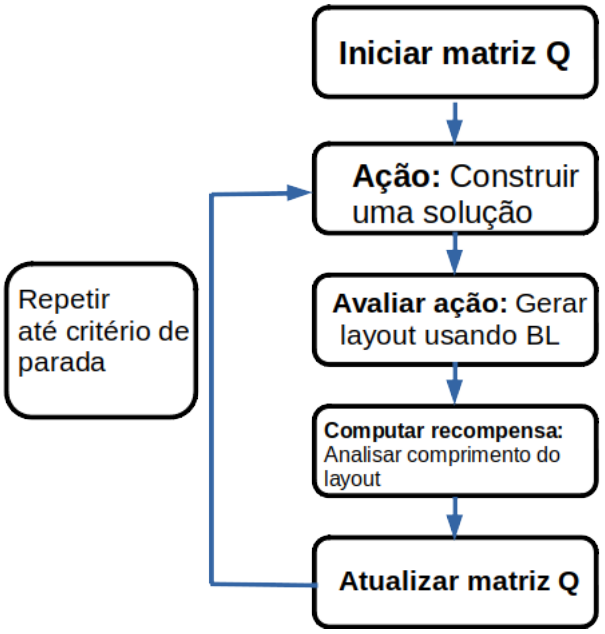


Figura 2: Passos do método de aprendizado por reforço.

e os cinco exemplares *rco(k)* ($k = 1, ..., 5$) baseados em [Oliveira et al., 2000]. Cada um dos exemplares está descrito na Tabela 1, onde na coluna "Exemplar"consta o nome de cada exemplar utilizado, a coluna "n"contém o número de tipos de peças distintos, a coluna "d"apresenta a multiplicidade (demanda) dos tipos de peça no exemplar e o número total de peças está descrito na coluna "m". Esses exemplares podem ser encontrados em [ESICUP, 2021].

Tabela 1: Descrição dos exemplares utilizados.

Exemplar	n	d	m
rco1	7	1	7
rco2	7	2	14
rco3	7	3	21
rco4	7	4	28
rco5	7	5	35
blazewicz1	7	1	7
blazewicz2	7	2	14
blazewicz3	7	3	21
blazewicz4	7	4	28
blazewicz5	7	5	35

Como já mencionado, os exemplares *rco* são compostos pelas envoltórias convexas das peças dos exemplares *blazewicz*. Na Figura 3, são ilustradas as peças de cada um dos conjuntos de exemplares.

Foram realizados dois grupos de testes: a primeira abordagem, denominada **R** (aprendizado por **R**eforço), está descrita no Algoritmo 1, enquanto a segunda abordagem, denominada **T** (**T**ransferência de aprendizado por reforço), é descrita no Algoritmo 2. De fato, para abordagem **T** a

Algoritmo 2 Aprendizado por reforço com transferência.

Dados de Entrada: Q, OW, BW, d, m, α e β
enquanto critério de parada não atingido **faça**
  procedimento ITERAÇÕES DE APRENDIZADO POR REFORÇO
    $S \leftarrow \emptyset$
    para cada posição da solução (j)
      $p_i \leftarrow e^{Q_{ij}},\ \forall i : d_i \geq 0$ ▷ Calcular contribuição de cada peça factível
      $S_j \leftarrow i$ ▷ Seleção por roleta ponderada
      $d_i \leftarrow d_i - 1$ ▷ Atualiza a demanda da peça selecionada
    $CW \leftarrow BL(S)$ ▷ Calcula o comprimento da solução usando a regra *bottom-left*
    se $(BW - CW) \geq 0$ ou $(OW - CW) \geq 0$ **então** ▷ Atualização da matriz Q
      para cada entrada da sequência
        $Q_{ij} \leftarrow Q_{ij} + \alpha$
    senão
      para cada entrada da sequência
        $Q_{ij} \leftarrow Q_{ij} + \beta$
    se $CW \leq BW$ **então** ▷ Atualização da melhor solução incumbente
      $BW \leftarrow CW$
  procedimento ITERAÇÕES COM SOLUÇÕES ALEATÓRIAS
    $S \leftarrow RS$ ▷ Cria uma solução factível aleatória com probabilidade uniforme
    $CW \leftarrow BL(S)$ ▷ Calcula o comprimento da solução usando a regra *bottom-left*
    se $(BW - CW) \geq 0$ ou $(OW - CW) \geq 0$ **então** ▷ Atualização da matriz Q
      para cada entrada da sequência
        $Q_{ij} \leftarrow Q_{ij} + \alpha$
    senão
      para cada entrada da sequência
        $Q_{ij} \leftarrow Q_{ij} - \beta$
    se $CW \leq BW$ **então** ▷ Atualização da melhor solução incumbente
      $BW \leftarrow CW$

matriz Q treinada para os exemplares *rco* foi transferida para os exemplares *blazewicz*, e vice-versa.

Os métodos de resolução foram implementados em Matlab 2020a e C++ e os testes computacionais foram realizados em um computador Intel® Core™ i5-7400 CPU 3.00GHz 64bits com 8Gb e sistema operacional Ubuntu 20.04. O critério de parada utilizado foi o tempo máximo de execução sendo que, para a abordagem **R**, o tempo máximo de execução foi definido como 700 segundos por exemplar e para a abordagem **T** esse tempo foi de 600 segundos para a criação da matriz Q durante o aprendizado por reforço mais 100 segundos para a execução da etapa de transferência de aprendizado (Algoritmo 2). O objetivo de utilizar um tempo de execução menor para **T** é que, como parte do aprendizado já é herdado (por meio da transferência de aprendizado) é esperado que um número menor de execuções seja necessário para a convergência do aprendizado.

Para a função de cálculo de recompensa os parâmetros α e β, utilizados nas Equações (1) e (2), foram tomados como $\alpha = 100$ e $\beta = 10 \times OW \times (CW - OW)$. Esses valores foram definidos por meio de testes computacionais preliminares considerando os intervalos $\alpha = [0, 500]$ e $\beta = \gamma \times OW \times (CW - OW)$, com $\gamma = [0, 50]$.

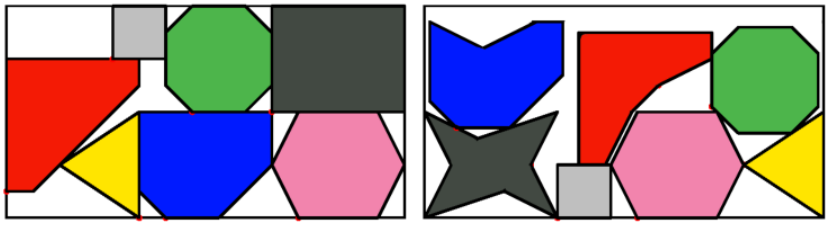


Figura 3: Peças dos exemplares *rco* (esquerda) e *blazewicz* (direita). As peças de mesma cor representam os pares de peças côncavas (*blazewicz*) e seus contornos convexos (*rco*).

Exemplares com peças não-convexas fazem com que o tempo de verificação de sobreposição entre peças aumente com o número de vértices da peça, reduzindo significantemente o número de execuções. A título de comparação, em 600 segundos, a abordagem de aprendizado por reforço faz 1024 execuções para o exemplar *rco*5 e somente 184 para o exemplar *blazewicz*5. Outra diferença entre os conjuntos de exemplares está na possibilidade de encaixe entre as peças. Devido às suas concavidades, exemplares *blazewicz* têm melhores opções de encaixe, o que permite obter soluções com comprimentos menores (Figura 3). Esse comportamento é observado quando são comparados os valores mínimos e máximos entre os exemplares *blazewicz* e *rco*. Logo, os limitantes OW utilizados são diferentes para cada exemplar (ver Tabela 4).

Na Tabela 4, são apresentados os comprimentos mínimos, máximos, a mediana e o desvio padrão obtidos pelas duas estratégias para os conjuntos de exemplares *rco* e *blazewicz*. Os melhores valores para mínimos, máximos e medianas estão destacados na tabela. A coluna "Ref."apresenta os valores mínimos encontrados para cada exemplar considerando o modelo exato de Cherri et al. [2016], implementado em linguagem C/C++ e resolvido utilizando o *software* de otimização ILOG CPLEX 12.6 e tempo execução de 3600s. A coluna "OW"contém os valores de limitante superior inicial para o comprimento da solução utilizados para os cálculos da recompensa.

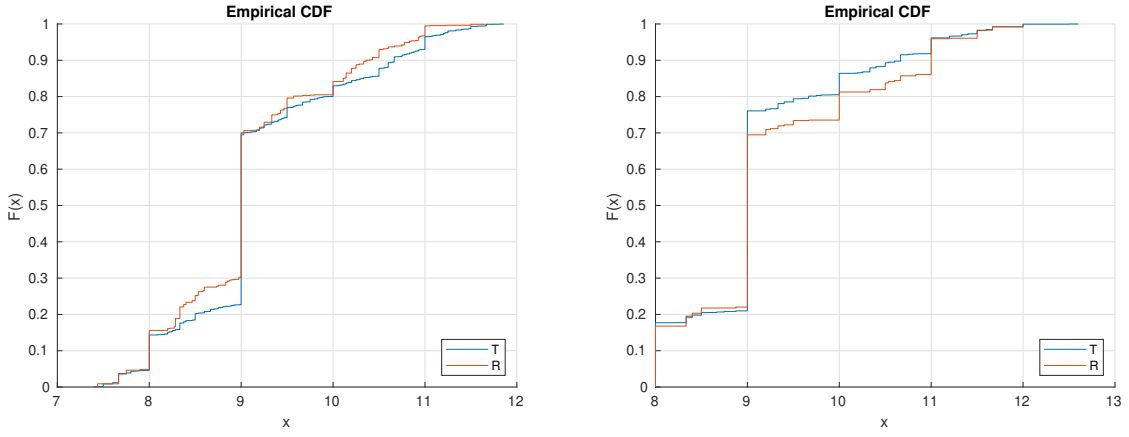
Tabela 2: Resultados obtidos pelos métodos de aprendizado por reforço com e sem transferência (**T e R**) de aprendizado para os exemplares *rco* e *blazewicz*.

			Mínimo		Máximo		Mediana		Desvio Padrão	
	OW	Ref.	**T**	**R**	**T**	**R**	**T**	**R**	**T**	**R**
rco1	8,00	8,00	8,00	8,00	12,00	12,00	9,00	9,00	0,89	0,99
rco2	17,00	14,87	15,33	15,50	22,33	20,66	17,50	17,00	1,00	0,83
rco3	25,00	22,44	23,00	22,33	32,00	32,00	26,00	26,00	1,21	1,21
rco4	29,00	30,44	31,00	30,00	39,66	40,00	34,00	34,00	1,27	1,29
rco5	41,00	38,40	39,11	37,66	47,40	49,00	42,00	42,11	1,38	1,39
blazewicz1	8,00	7,40	7,40	7,43	11,85	11,50	9,00	9,00	0,87	0,85
blazewicz2	16,00	14,25	14,58	14,50	20,80	21,38	17,08	17,03	1,08	1,06
blazewicz3	22,00	21,68	21,86	22,13	30,58	29,41	24,84	24,95	1,33	1,19
blazewicz4	29,00	29,44	30,19	30,50	35,43	37,42	32,75	32,94	1,58	1,49
blazewicz5	36,00	41,05	37,36	37,47	43,97	44,81	40,27	40,15	1,79	1,79

Para os exemplares *rco*, a abordagem usando aprendizado por reforço (**R**) obteve os melhores comprimentos mínimos quando comparado com a abordagem **T**, exceto para o exemplar *rco2*. Contudo para os exemplares *blazewicz*, o comportamento oposto é observado, exceto para o exemplar *blazewicz2*. Esse comportamento se justifica pela qualidade da matriz Q herdada pelos

exemplares *blazewicz*, qualidade que se deve ao alto número de execuções realizada para os exemplares *rco*. O desempenho superior da abordagem **T** também é evidenciado nas análises do valor máximo. Contudo, para as duas abordagens, nota-se a perda de qualidade na solução conforme a dimensão dos exemplares aumenta. Essa perda está relacionada ao número de execuções possíveis dentro do tempo limite. Vale destacar que para os exemplares *rco1* e *blazewicz1* as soluções ótimas foram obtidas.

A fim de comparar o desempenho das abordagens foi utilizado o teste de Kolmogorov-Smirnov biamostral. Os experimentos computacionais foram realizados usando a função *kstest2* em Matlab 2020a. O teste de Kolmogorov-Smirnov biamostral mostrou diferença estatística entre as abordagens **R** e **T** para os exemplares *blazewicz1*, *rco1* e *rco2*. Para ilustrar a diferença de desempenho entre as abordagens, nas Figuras 4 e 5, é apresentada a frequência acumulada (CDF) das soluções encontradas no decorrer do tempo de execução. Vale ressaltar que o método de Kolmogorov-Smirnov é baseado na comparação da CDF gerada para cada uma das abordagens. Para essas figuras, o eixo vertical representa a porcentagem de soluções de comprimento menor ou igual a determinado valor (eixo horizontal). Intervalos da curva com crescimento estritamente vertical sinalizam a convergência do método para aquele determinado valor, por exemplo, na Figura 4(a) esse comportamento pode ser visto para o comprimento 9.



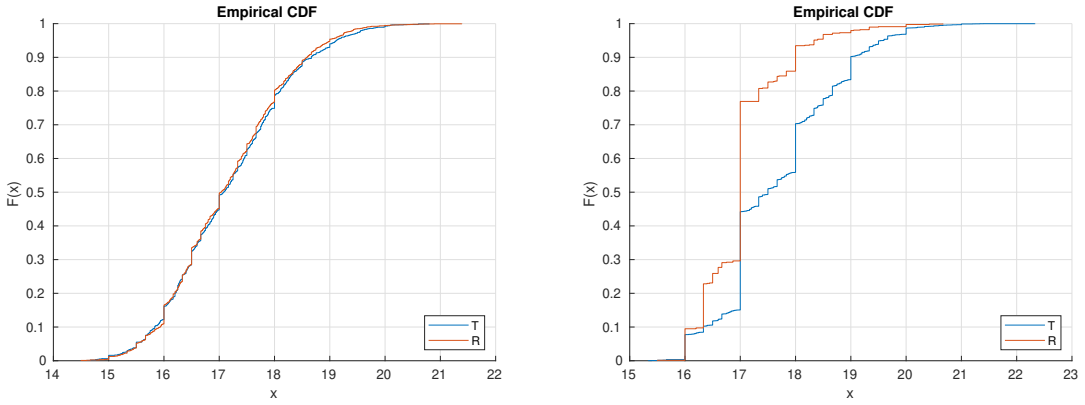
(a) blazewicz1. (b) rco1.

Figura 4: Comparação dos comprimentos gerados para os exemplares com 7 peças.

Enquanto que para os exemplares *blazewicz1*, *rco1* e *rco2* (Figuras 4(a), 4(b) e 5(b), respectivamente) é possível observar a convergência das duas abordagens de solução, para o exemplar *blazewicz2* (Figura 5(a)) isso já não acontece. O aumento no número de combinações possíveis entre as peças e a redução no número de execuções dentro do tempo limite são os fatores decisivos para esse desempenho, de fato, esse comportamento se acentua com o aumento da multiplicidade das peças.

5. Conclusões

Nesse trabalho, aplicamos duas técnicas de aprendizado de máquina na resolução de problemas de corte e empacotamento de peças irregulares: aprendizado por reforço e transferência de aprendizado. Foram selecionados dois conjuntos de exemplares para análise, de forma que um conjunto era composto por peças convexas e não-convexas (*blazewicz*) e o segundo conjunto era composto pela envoltória convexa das peças do primeiro conjunto de exemplares (*rco*). Computacionalmente, peças irregulares não-convexas demandam maior esforço para gerar soluções sem



(a) blazewicz2. (b) rco2.

Figura 5: Comparação dos comprimentos gerados para os exemplares com 14 peças.

sobreposição, fazendo com que o número de avaliações durante o processo de aprendizado seja significativamente reduzido. Logo, a transferência de aprendizado de exemplares com peças convexas para exemplares com peças não-convexas mostrou-se vantajoso. Testes estatísticos mostraram que o método com transferência de aprendizado atinge um desempenho similar ao método de aprendizado por reforço mesmo com tempo de execução inferior. Além disso, à medida que a dimensão dos exemplares cresce, o método com transferência de aprendizado apresentou soluções com melhores valores mínimos de comprimento. Em pesquisas futuras, pretendemos expandir os exemplares analisados e avaliar a possibilidade de iniciar o treinamento com peças convexas e depois dar continuidade utilizando a forma original das peças.

Agradecimentos

As autoras agradecem à Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) (processos n. 2020/15707-6, 2018/07240-0 e 2013/07375-0) e ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) (processo n. 308761/2018-9). Elas também agradecem ao Laboratório de Otimização do ICMC/USP (LOt - `www.otm.icmc.usp.br`) e aos revisores deste trabalho.

Referências

Baker, B. S., Coffman, E. G., Jr, e Rivest, R. L. (1980). Orthogonal packings in two dimensions. SIAM Journal on computing, 9(4):846–855.

Bengio, Y., Lodi, A., e Prouvost, A. (2020). Machine learning for combinatorial optimization: a methodological tour d'horizon. European Journal of Operational Research, 290(2):405–421.

Bennell, J. A. e Oliveira, J. F. (2009). A tutorial in irregular shape packing problems. Journal of the Operational Research Society, 60:S93–S105.

Bennell, J. e Dowsland, K. (2001). Hybridising tabu search with optimisation techniques for irregular stock cutting. Management Science, 47(8):1160–1172.

Błażewicz, J., Hawryluk, P., e Walkowiak, R. (1993). Using a tabu search approach for solving the two-dimensional irregular cutting problem. Annals of Operations Research, 41(4):313–325.

Cherri, L. H., Mundim, L. R., Andretta, M., Toledo, F. M., Oliveira, J. F., e Carravilla, M. A. (2016). Robust mixed-integer linear programming models for the irregular strip packing problem. European Journal of Operational Research, 253(3):570–583.

Elkeran, A. (2013). A new approach for sheet nesting problem using guided cuckoo search and pairwise clustering. European Journal of Operational Research, 231(3):757–769.

ESICUP (2021). Working group on cutting and packing within EURO. URL www.euro-online.org/websites/esicup/data-sets/#1535972088237-bbcb74e3-b507.Acesso:16/05/2021.

Gahm, C., Uzunoglu, A., Wahl, S., Ganschinietz, C., e Tuma, A. (2021). Applying machine learning for the anticipation of complex nesting solutions in hierarchical production planning. European Journal of Operational Research.

Gomes, A. e Oliveira, J. F. (2006). Solving irregular strip packing problems by hybridising simulated annealing and linear programming. European Journal of Operational Research, 171(3):811–829.

Gomes, A. e Oliveira, J. F. (2002). A 2-exchange heuristic for nesting problems. European Journal of Operational Research, 141(2):359–370.

Hu, H., Zhang, X., Yan, X., Wang, L., e Xu, Y. (2017). Solving a new 3D bin packing problem with deep reinforcement learning method. arXiv preprint arXiv:1708.05930, p. 1–7.

Oliveira, J. F., Gomes, A. M., e Ferreira, J. S. (2000). Topos–a new constructive algorithm for nesting problems. OR-Spektrum, 22(2):263–284.

Powell, W. B. (2016). Perspectives of approximate dynamic programming. Annals of Operations Research, 241(1):319–356.

Rakotonirainy, R. G. (2020). A machine learning approach for automated strip packing algorithm selection. ORiON, 36(2):73–88.

Sato, A. K., Martins, T. C., Gomes, A. M., e Tsuzuki, M. S. G. (2019). Raster penetration map applied to the irregular packing problem. European Journal of Operational Research, 279(2):657–671.

Watkins, C. J. e Dayan, P. (1992). Q-learning. Machine learning, 8(3-4):279–292.

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.