Análise e Abordagens Multiobjetivo para o Problema de Escalonamento de Pátio Operado por Guindastes

Thomás de A. F. Gouvêa
Programa de Pós-Graduação em Ciência da Computação, Universidade Federal de Ouro Preto
Campus Universitário Morro do Cruzeiro, s/n, Ouro Preto/MG - 35400-000
thomasafg@hotmail.com

Rodrigo Silva
Departamento de Computação, Universidade Federal de Ouro Preto
Campus Universitário Morro do Cruzeiro, s/n, Ouro Preto/MG - 35400-000
rodrigo.silva@ufop.edu.br

Túlio A. M. Toffolo
Departamento de Computação, Universidade Federal de Ouro Preto
Campus Universitário Morro do Cruzeiro, s/n, Ouro Preto/MG - 35400-000
tulio@toffolo.com.br

RESUMO

Pátios operados por pontes rolantes são utilizados por diversas empresas, tendo fundamental importância em suas operações logísticas. Neste contexto, este trabalho aborda o Problema de Escalonamento de Pátio Operado por Guindastes, que consiste em definir onde estocar cada produto no pátio e por meio de qual guindaste. O objetivo é minimizar tanto os atrasos na entrega e no depósito de produtos quanto os conflitos entre produtos alocados. Neste trabalho é demonstrado que este dois objetivos são conflitantes e uma heurística multiobjetivo é proposta para encontrar um conjunto de soluções não dominadas. A heurística resultou em conjuntos não-dominados superiores aos da abordagem baseada em NSGA-2. Apesar de considerar a versão multiobjetivo do problema, a heurística foi capaz de melhorar a solução de 52 instâncias da versão mono-objetivo do problema disponíveis na literatura.

PALAVRAS CHAVE. Otimização Multiobjetivo, Problema de Escalonamento de Pátio Operado por Guindastes, Alocação de Produtos, Escalonamento de Guindastes.

ABSTRACT

Crane-operated warehouses are employed by several companies, having critical influence upon their logistic operations. In this context, this work addresses the Crane-operated Warehouses Scheduling Problem, which consists in defining where to store each product in the yard and which crane should be used to transport it. The objective is to minimize both product delivery delays and conflicts among stored products. This work demonstrates the multiobjective nature of the problem and proposes a multiobjective heuristic to generate a set of non-dominated solutions. The heuristic resulted in better non-dominated sets than those obtained by the approach based on NSGA-2. Moreover, despite considering the multiobjective version of the problem, the heuristic was capable of improving the best known solution for 52 instances of the mono-objective version of the problem available in the literature.

KEYWORDS. Multiobjective Optimization, Crane-operated Warehouse Scheduling Problem, Location Assignment, Crane Scheduling

1. Introdução

Pátios operados por guindastes ou pontes rolantes são utilizados para armazenamento de grandes cargas e materiais, sendo cruciais nas indústrias portuária e pesada (siderurgia e metalurgia). Seu controle é efetuado de forma segura e eficaz à distância, porém há limitações quanto aos pontos de entrada e saída de produtos, localizados nas laterais, e quanto ao acesso aos produtos pelo guincho, restrito aos produtos no topo das pilhas. Devido a essas limitações, diversos estudos (e.g. Boysen e Emde [2016], Ku e Arthanari [2016]) foram realizados de forma a reduzir o deslocamento do guindaste e operações de *reshuffle*, que ocorrem quando é necessário realocar produtos para acessar outros depositados abaixo destes.

Nos trabalhos citados anteriormente, apenas um guindaste opera a movimentação de produtos nos pátios, porém, na prática, a maioria dos pátios operam com dois ou mais guindastes. Assim, é necessário definir com cautela qual guindaste será responsável por cada requisição a fim de evitar colisões durante as movimentações, uma vez que neste modelo os guindastes compartilham o mesmo trilho. Li et al. [2009], Li et al. [2012] e Wu et al. [2015] propuseram modelos exatos e heurísticas baseadas em tempo discreto e contínuo para reduzir a movimentação dos guindastes.

Heshmati et al. [2019] propuseram uma formulação alternativa para o Problema de Escalonamento de Pátios Operados por Guindastes, denominado *Crane-operated Warehouse Scheduling Problem* (CWSP), para pátios nos quais os produtos são conflitantes entre si. Por exemplo, em pátios de descanso em siderúrgicas, onde produtos que acabaram de sair de algum processo e, portanto, têm alta temperatura e precisam ser depositados para resfriamento. A taxa de resfriamento de um produto é alterada por aqueles que estão à sua volta, sendo que a alteração desta taxa influencia na qualidade do produto final. Assim, se o depósito de um produto é designado para a vizinhança de um produto conflitante, um valor previamente estimado é adicionado à função objetivo. Para controlar a importância relativa entre o escalonamento de guindastes e os conflitos entre produtos, Heshmati et al. [2019] utilizaram um par de pesos e propuseram um modelo exato de programação linear e uma heurística baseada em *Late Acceptance Hill Climbing* [Burke e Bykov, 2017] para abordar o problema.

Segundo Arroyo [2002], o método de aplicar pesos a cada objetivo de um problema e transformá-lo em mono-objetivo é uma forma simples para obter soluções de um problema multiobjetivo. Contudo, é difícil determinar estes pesos a priori, principalmente quando se considera objetivos distintos, como no CWSP, onde compara-se conflito de alocação com tempo de atraso de pedidos. Assim, este estudo propõe uma heurística para o CWSP, e a análise desta como solução para o problema multiobjetivo.

2. Problema de Escalonamento de Pátio Operado por Guindastes

Antes de definir o problema, é necessário analisar o *layout* do pátio, equipamentos, disposição dos produtos e ritmo de trabalho. Nos pátios operados por pontes rolantes ou guindastes, os produtos entram e saem do pátio pelas laterais deste, podendo estes pontos serem somente de entrada, somente de saída ou entrada/saída. Também é comum nestes pátios os produtos serem empilhados para melhor aproveitamento do espaço disponível.

As pontes rolantes ou guindastes são os equipamentos que manipulam os produtos dentro do pátio. Elas se movem sobre trilhos ou vigas suspensas percorrendo o pátio em um eixo. Um guincho acoplado à ponte permite o movimento ao eixo perpendicular e garante assim o acesso a todo o espaço dentro do pátio. A Figura 1 exemplifica um pátio operado por pontes rolantes, em que as setas indicam o movimento destas e como elas conseguem acessar todo o pátio na prática.

Os produtos armazenados no pátio serão manipulados, geralmente, apenas quando se deseja depositar ou retirar um produto. Estas atividades são denominadas requisições e devem ser avaliadas pelo operador do pátio para garantir os atendimentos destas no prazo. Cada requisição tem um horário de liberação, seja o horário em que o produto chegou no pátio ou prazo de armazenagem que o produto deveria ficar dentro do pátio, e, portanto, o produto não pode ser manipulado antes deste tempo. Também há um prazo de entrega, crucial para a cadeia logística a qual este

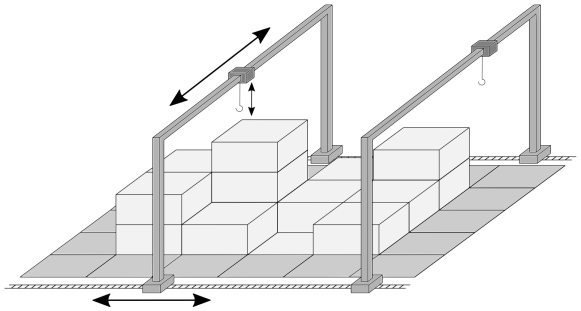
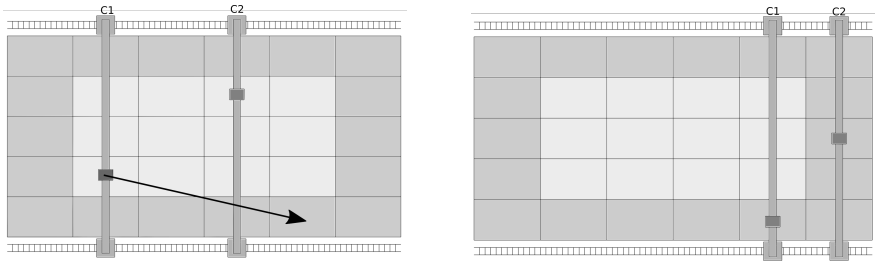


Figura 1: Esboço de um pátio operado por pontes rolantes.

pátio pertence. A eficiência do pátio é medida de acordo com a pontualidade em que ele entrega os produtos.

Para maior controle, muitos pátios tem as requisições de um período conhecidas, ou seja, o operador sabe com antecedência quantas requisições de entrada e saída serão executadas, seus horários de liberação e entrega, locais de chegada e saída dos produtos. Assim, o operador tem a informação necessária para decidir como manipular os produtos para atender as requisições daquele período. Em um pátio com apenas uma máquina e sem empilhamento, o operador precisa apenas garantir que a máquina percorra o menor caminho para atender as requisições e assim ter a máxima eficiência. Porém, a maioria dos pátios não possui este desenho. Em geral, eles possuem dois ou mais guindastes que compartilham o mesmo trilho, vários produtos para empilhar, limitações de movimentação, diferenças de velocidades e de operação dos guindastes, entre outros complicadores. Destas características há duas em especial que ocorrem na maioria dos pátios e que são determinantes para a eficiência dos mesmos: presença de produtos empilhados e dois ou mais guindastes em operação.

A presença de produtos empilhados no pátio é um complicador, pois quando há uma requisição de saída de produto que esteja abaixo de outro, deve-se primeiro retirar o produto acima. Este movimento, chamado de *reshuffle*, ocupa uma máquina por certo tempo, portando os operadores sempre desejam evitá-los. A operação de dois ou mais guindastes é outro complicador, pois toda a movimentação deve garantir que não haja colisão entre os guindastes. Por exemplo, imagine que um guindaste deseja transportar um produto para um ponto além de outro guindaste, como C1 na Figura 2(a). O guindaste C2 deve se movimentar também (Figura 2(b)) para evitar a colisão, mantendo uma distância mínima de segurança entre eles.



(a) (b)

Figura 2: Exemplo de movimentação de guindastes (Vista superior).

Definidas as operações comuns em pátios operados por pontes rolantes, pode-se definir formalmente os dois problemas que compõem o CWSP, são eles: (i) o Problema de Escalonamento de Guindastes, do inglês *Crane Scheduling Problem* (CSP), e (ii) o Problema de Alocação de Produtos, do inglês *Location Assignment Problem* (LAP).

O CSP consiste em decidir quando e qual guindaste irá atender cada requisição de entrada,

saída ou *reshuffle*, com o objetivo de reduzir o atraso no atendimento destas, respeitando-se todas as restrições de locomoção dos guindastes e disponibilidade dos produtos. A Equação (1) define a função objetivo deste problema de minimização, sendo $\boldsymbol{R}$ o conjunto de requisições, t_r^f o tempo de atendimento da requisição r e t_r^e o prazo de entrega da mesma.

$$f(t^f) = \text{min.} \sum_{r \in \boldsymbol{R}} \max(0, t_r^f - t_r^e) \tag{1}$$

O LAP consiste em definir a posição dos produtos no pátio de forma a minimizar o conflito entre produtos vizinhos. Sua função objetivo é definida pela Equação (2), onde: (i) $\boldsymbol{R^I}$ é o conjunto de todas as requisições de entrada e *reshuffle*; (ii) $\boldsymbol{D}$ é o conjunto de locais disponíveis para depósito; (iii) $x_{r,d}$ é variável que indica que o produto referente à requisição r foi depositado no local d; (iv) $\gamma_{r,d}$ representa o conflito gerado pela requisição r e os produtos vizinhos ao local d; (v) $y_{r,p}$ é a variável que indica que o produto referente à requisição r foi depositada na vizinhança do produto referente à requisição p; e (vi) $\omega_{r,p}$ representa o conflito entre produtos de duas requisições r e p.

$$f(x, y) = \text{min.} \sum_{r \in \boldsymbol{R^I}} \sum_{d \in \boldsymbol{D}} \gamma_{r,d} x_{r,d} + \sum_{r \in \boldsymbol{R^I}} \sum_{p \in \boldsymbol{R^I}} \omega_{r,p} y_{r,p} \tag{2}$$

A integração do LAP e do CSP resulta no CWSP, que considera os dois objetivos: (i) minimizar o atraso dos pedidos e (ii) minimizar o custo referente aos conflitos entre produtos vizinhos no pátio. Portanto, o CWSP contempla os objetivos de ambos LAP e CSP, dados pelas Equações (3)–(5).

$$\text{minimizar} \quad \mathbf{f}(\mathbf{x}) = [h(x, y), g(t^f)] \tag{3}$$

$$h(x, y) = \sum_{r \in \boldsymbol{R^I}} \sum_{d \in \boldsymbol{D}} \gamma_{r,d} x_{r,d} + \sum_{r \in \boldsymbol{R^I}} \sum_{p \in \boldsymbol{R^I}} \omega_{r,p} y_{r,p} \tag{4}$$

$$g(t^f) = \sum_{r \in \boldsymbol{R}} \max(0, t_r^f - t_r^e) \tag{5}$$

Embora o problema tenha dois objetivos bem definidos, uma abordagem multiobjetivo só se justifica se houver conflito entre eles, ou seja, se houver uma relação de compromisso entre os objetivos. Assim, na próxima seção, o conflito entre os objetivos é avaliado para várias instâncias.

3. Análise de conflito entre o CSP e LAP

Dado um conjunto de soluções $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_m\}$, o conflito entre dois objetivos, f_1 e f_2, pode ser estimado aplicando-se a seguinte fórmula:

$$\text{conflito}(f_1, f_2, \mathcal{X}) = \frac{-(\rho(\mathcal{F}_1(\mathcal{X}), \mathcal{F}_2(\mathcal{X})) - 1)}{2}$$

em que $\mathcal{F}_i(\mathcal{X}) = \{f_i(\mathbf{x}_1), f_i(\mathbf{x}_2), ..., f_i(\mathbf{x}_m)\}$ e $\rho(\mathcal{F}_1(\mathcal{X}), \mathcal{F}_2(\mathcal{X}))$ é a correlação de Spearman entre os conjuntos $\mathcal{F}_1(\mathcal{X})$ e $\mathcal{F}_2(\mathcal{X})$.

O conflito assume um valor entre 0 e 1. Ele é máximo, igual a 1, quando sempre que uma solução melhora em um objetivo ela piora no outro. O conflito é mínimo, igual a 0, quando não existe compromisso entre os objetivos. Ou seja, sempre que uma solução melhora para um objetivo, ela também melhora para o outro objetivo. Para uma descrição mais detalhada sobre conflito, veja Freitas et al. [2013].

Para avaliar o conflito entre o CSP e o LAP, instâncias de Heshmati et al. [2019][1] foram selecionadas. Estas instâncias recebem a nomenclatura RT_P_O, onde:

[1]Disponível em: `https://bitbucket.org/Sam-Hes/cwsp.git` - Acessado em 27 de Abril de 2020

- R é o **Número de requisições**, que pode assumir os valores 10, 20, 30, 50 ou 70 requisições de entrada e saída;
- T é o **Tamanho do pátio**, que assume os valores s (10×25, totalizando 250 locais de depósito), m (15×35, ou 525 locais de depósito) ou l (20×50, 1000 locais de depósito);
- P é o **Tamanho máximo da pilha de produtos**, que pode assumir os valores 1, 3 ou 5;
- O é a **Porcentagem de ocupação do pátio**, que pode assumir os valores de 30%, 50% ou 70% do pátio ocupado por produtos.

As outras informações foram geradas aleatoriamente sendo que: os tempos de liberação seguem uma distribuição Poisson com média 26.5, os prazos limite seguem uma distribuição logarítmica normal com média 0.0666 vezes o tamanho do pátio e desvio padrão de 0.1 vezes o comprimento do pátio, os produtos já alocados seguem uma distribuição uniforme entre 0 e o total de locais de depósito, e seus valores de penalização também recebem uma distribuição uniforme entre 0 e 50.

A Figura 3 mostra os valores de conflito para um conjunto de soluções aleatórias. Pode-se verificar que, em todos os casos, o conflito está entre 0.4 e 0.6 e varia pouco entre as instâncias. Isso indica que estes objetivos são conflitantes, independentemente da configuração do problema, o que reforça a necessidade de utilização de um método multiobjetivo.

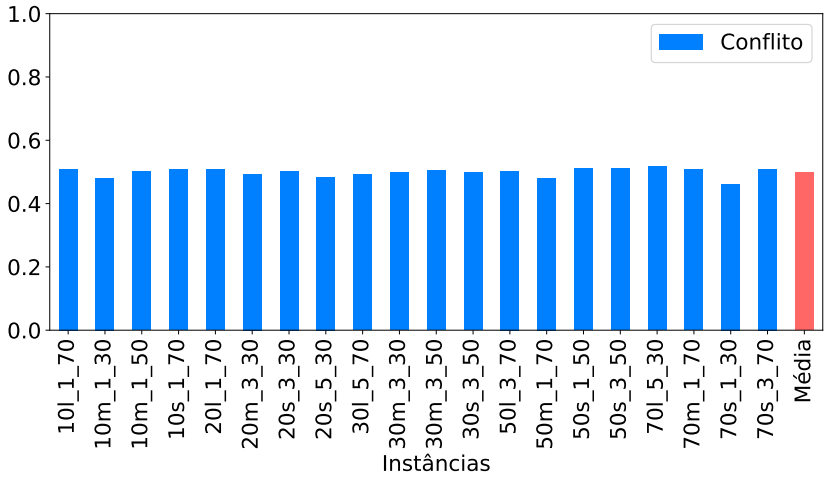


Figura 3: Conflito entre objetivos para soluções geradas aleatoriamente. Em vermelho, a médias dos conflitos de todas as instâncias.

Da mesma forma, se observar um gráfico de coordenadas paralelas para um destes conjuntos de soluções, vide Figura 4, é possível perceber que algumas soluções são ruins para o primeiro objetivo e boas para o segundo, entretanto algumas são boas ou ruins para ambos. Esta dificuldade em se determinar a relação entre os objetivos também reforça a necessidade de aplicação de um método multiobjetivo para este problema.

4. Otimização Multiobjetivo

Para uma abordagem multiobjetivo é necessário entender o conceito de Dominância e Pareto-otimalidade. Diferente de problemas mono-objetivo, o espaço de soluções de problemas multiobjetivo é definido em várias dimensões, dificultando a avaliação das soluções. Neste casos, utiliza-se o conceito de Pareto-dominância para comparar soluções.

Considerando um problema em que todas as funções objetivo, $f_k, k = 1, ..., m$, são de minimização, dizemos que $\mathbf{x}_i$ "domina" $\mathbf{x}_j$, se $f_k(\mathbf{x}_i) \leq f_k(\mathbf{x}_j)$ para todo k, e $f_k(\mathbf{x}_i) < f_k(\mathbf{x}_j)$

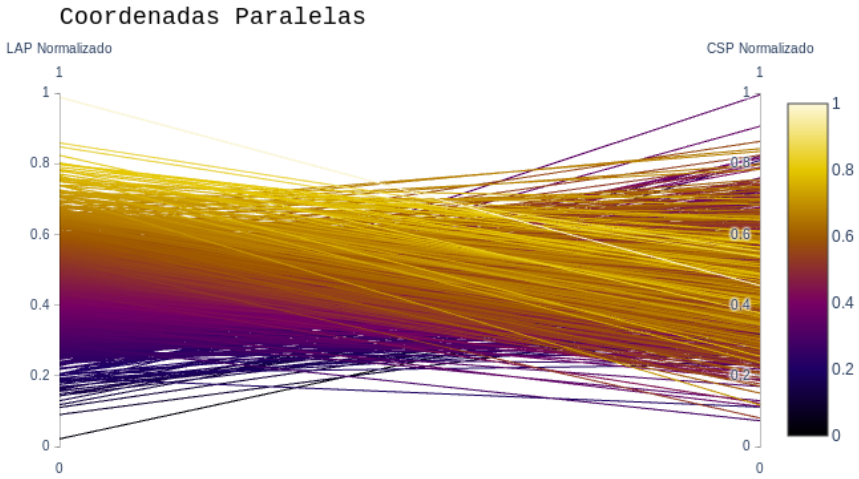


Figura 4: Gráfico de coordenadas paralelas das soluções geradas aleatoriamente para a instância 20s_3_30. Cada linha representa uma solução e linhas se cruzando indicam conflito.

para algum k. Dizemos que $\mathbf{x}_i$ "é dominada" por $\mathbf{x}_j$, se $f_k(\mathbf{x}_i) \geq f_k(\mathbf{x}_j)$ para todo k, e $f_k(\mathbf{x}_i) > f_k(\mathbf{x}_j)$ para algum k. Nos demais casos, $\mathbf{x}_i$ é "indiferente" à $\mathbf{x}_j$.

Se analisadas todas as soluções possíveis de um problema, as soluções que não são "dominadas" por nenhuma outra são consideradas, igualmente, as melhores soluções para o problema, sendo este conjunto denominado de Fronteira Pareto-ótima ou soluções Pareto-ótimas. Portanto, em um problema multiobjetivo não se busca uma única solução ótima e sim um conjunto de soluções Pareto-ótimas. Na próxima seção é descrito o algoritmo proposto neste trabalho, que atende esta necessidade de uma abordagem multiobjetivo.

5. Algoritmo GLS+ILS

Antes de iniciar a descrição do algoritmo é necessário descrever quais informações estão disponíveis e como a solução é representada. As informações para o problema são divididas em informações sobre os produtos, locais de depósito e guindastes. Para o primeiro, tem-se quais produtos irão entrar no pátio, sair ou serão movimentados como *reshuffle* ($\boldsymbol{R}$), os prazos de liberação (t^l) e entrega (t^e) e a penalidade de se alocar cada produto ao lado de outro (ω). Sobre os locais de depósito, tem-se onde cada produto já está alocado, de onde virão os produtos de entrada (o), para onde irão os produtos de saída (d) e quantos produtos podem ser empilhados no pátio. Para os guindastes, é conhecida a quantidade de guindastes no pátio e a distância mínima de segurança (ts) que eles devem manter entre si.

Uma representação *indireta* da solução é utilizada. A Figura 5 ilustra esta representação, que consiste em um vetor ordenado contendo todas as requisições. Para cada requisição são associados o guindaste que fará a movimentação e o local de destino do produto. Uma solução indireta pode ser decodificada em uma solução completa, contendo todos os tempos de início e fim de cada movimentação, utilizando o Algoritmo 1. Note que as restrições de movimentação de cada guindaste são utilizadas para calcular o tempo de início de cada requisição. Esta representação indireta da solução simplifica a avaliação de vizinhos e permite a utilização das vizinhanças apresentadas na próxima seção.

5.1. Vizinhanças

Diversas vizinhanças foram definidas para explorar o espaço de soluções do problema, todas considerando apenas soluções viáveis. Dada uma solução inicial $\mathbf{x}$, soluções vizinhas são obtidas a partir de movimentos que consistem em alterar parcialmente esta solução. Como uma solução é composta por três componentes (destino, guindaste associado e ordem) de cada requisição, os movimentos que definem as vizinhanças visam alterar um ou mais destes componentes. Ao todo seis vizinhanças são consideradas, definidas pelos seguintes movimentos:

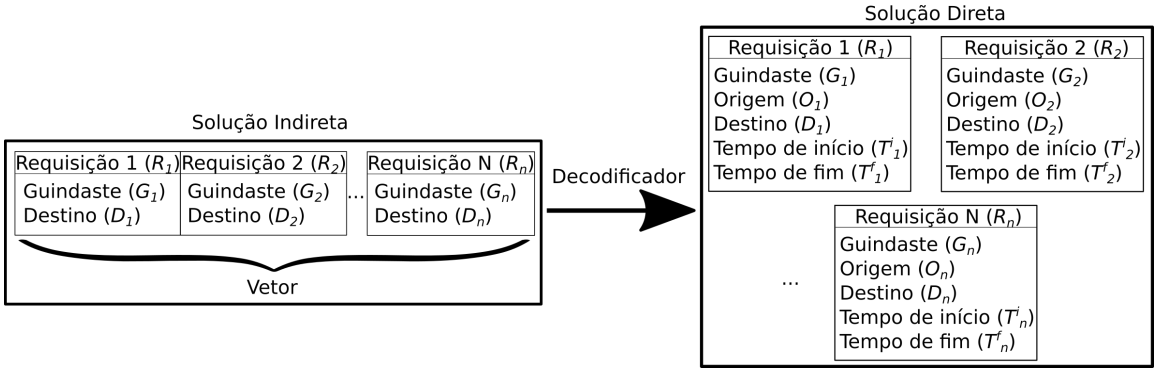


Figura 5: Representações das soluções indireta e direta.

Algoritmo 1: Decodificador da solução indireta

Entrada: Vetor de requisições R
Saída: Tempo de conclusão t^f para cada requisição em R

$E_g \leftarrow \emptyset \quad \forall g \in G$
para $r \in R$ **faça**
  $t^i_r \quad t^l_r$
  $g \quad G_r$
  se E_g *não está vazia* **então**
    q última requisição em E_g
    $t^i_r \quad max(t^i_r, t^i_q + tm_{o_q d_q} + tm_{d_q o_r})$
  para $h \in G, h \neq G_r$ **faça**
    E_h^{-1} inverso de E_h
    para $p \in E_h^{-1}$ **faça**
      se r *e* p *podem colidir* **então**
        se (d_p *está à direita* o_r *e* $h < g$) *ou* (d_p *está à esquerda* o_r *e* $h > g$) **então**
          $t^i_r \leftarrow max(t^i_r, t^i_p + tm_{o_p d_p} + tm_{d_p o_r} + ts_{gh})$
        se (d_p *está à esquerda* o_r, d_p *está à direita* o_p *e* $h < g$) *ou* (d_p *está à direita* o_r, d_p *está à esquerda* o_p *e* $h > g$) **então**
          $t^i_r \leftarrow max(t^i_r, t^i_p + tm_{o_p d_p} - tm_{d_p o_r} + ts_{gh})$
        se (d_p *está à esquerda* o_r, d_p *está à esquerda* o_p *e* $h < g$) *ou* (d_p *está à direita* o_r, d_p *está à direita* o_p *e* $h > g$) **então**
          $t^i_r \leftarrow max(t^i_r, t^i_p + tm_{o_p o_r} + ts_{gh})$
        encerrar este loop
  $t^f_r \leftarrow t^i_r + tm_{o_i d_i}$
  Inserir a requisição r em E_g
return

- Destino aleatório (Figura 6(a)): um novo destino é escolhido para uma requisição aleatória;
- Destino mais próximo à origem (Figura 6(b)): um novo destino aleatório é escolhido para uma requisição aleatória, porém este destino deve ser mais próximo do local de origem desta requisição que o destino anterior;
- Destino vizinho com menor custo (Figura 6(c)): um dos vizinhos do destino anterior é escolhido como novo destino para uma requisição aleatória se tiver menor custo de alocação;
- Guindaste aleatório (Figura 6(d)): outro guindaste é designado para uma requisição aleatória;
- Troca de ordem de uma requisição (Figura 6(e)): uma requisição aleatória é retirada da ordem das requisições e inserida em outro posição;

- Melhor ordem de uma janela de requisições (Figura 6(f)): uma janela de quatro requisições em sequência são selecionadas e escolhida a melhor permutação entre elas.

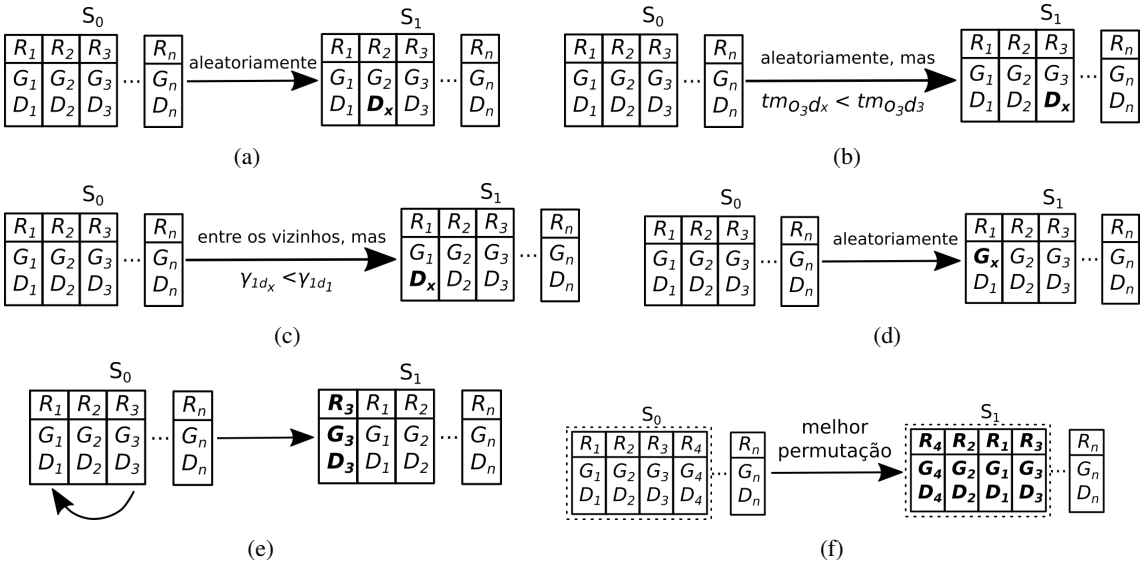


Figura 6: Estruturas de Vizinhança.

5.2. Heurística

Diferente de um problema mono-objetivo, onde a heurística deve fazer a busca por apenas uma solução ótima, este problema, que possui características multiobjetivo, deve ser representado por um conjunto de soluções ótimas, denominada Fronteira de Pareto. Para isso, são descritos a seguir o Algoritmo 2, uma combinação das meta-heurísticas Guided Local Search (GLS) (Alsheddy e Tsang [2010]) e Iterated Local Search (ILS) (Lourenço et al. [2003]), proposto neste trabalho para encontrar uma aproximação da Fronteira de Pareto, e o Algoritmo 3, conhecido como Non-dominated Sorting Genetic Algorithm (NSGA-2) (Deb et al. [2002]), comumente utilizado para problemas mutliobjetivo e utilizado para comparação com o Algoritmo 2 proposto.

No Algoritmo 2, a ideia principal é, partindo de uma solução inicial aleatória, realizar várias buscas locais adicionando as soluções encontradas em cada busca local no conjunto de soluções não dominadas, filtrando o conjunto para retirar soluções dominadas e manter a diversidade de soluções através da métrica de *crowding distance*. Esta métrica é descrita pela Equação (6), em que $f_k(\mathbf{x})$ é o valor do $k-$ésimo objetivo da solução $\mathbf{x}$.

$$\text{crowding distance}(\mathbf{x}_i) = \sum_{k=1,\ldots,m} \begin{cases} \infty, & \text{se } f_k(\mathbf{x}_i) = f_{k,max} \text{ ou } f_{k,min} \\ \frac{f_k(\mathbf{x}_{i+1}) - f_k(\mathbf{x}_{i-1})}{f_{k,max} - f_{k,min}} \end{cases} \tag{6}$$

Particularizando o Algoritmo 2, o GLS é utilizado por seu princípio de alteração de peso dos objetivos para escapar de ótimos locais. Assim as funções objetivo do problema são combinadas com a variável λ, como mostrado Equação (7). Já o ILS é utilizado no ciclo interior do GLS para uma busca mais profunda, mantendo a melhor solução encontrada com o λ inicial como guia. A Pertubação e a Busca Local do ILS são feitas utilizando as estruturas de vizinhança descritas na Secção 5.1. Na Busca Local uma das estruturas vizinhanças é selecionada aleatoriamente. Um movimento é aplicado caso a solução permaneça viável e em caso de melhora, utilizando o λ modificado pelo GLS. Este processo é repetido por até M iterações sem melhora. Na Pertubação, um movimento de uma das estruturas de vizinhanças, selecionada aleatoriamente, é aplicado independente do seu valor. O processo é repetido quantas vezes for o valor do $nivel$.

$$f_\lambda(x, y, t^f) = \lambda \left(\sum_{r \in \boldsymbol{R^I}} \sum_{d \in \boldsymbol{D}} \gamma_{r,d} x_{r,d} + \sum_{r \in \boldsymbol{R^I}} \sum_{p \in \boldsymbol{R^I}} \omega_{r,p} y_{r,p} \right) + (1 - \lambda) \sum_{r \in \boldsymbol{R}} \max \left(0, t_r^f - t_r^e\right) \tag{7}$$

Algoritmo 2: GLS + ILS

Entrada: Solução inicial S_0, λ_0 inicial, tempo máximo de execução $tempo^{max}$, número máximo de soluções Pareto c^{max}, número de iterações sem melhora M
Saída: Conjunto de soluções Pareto C.

```
1  S ← S_0
2  C ← ∅
3  λ ← λ_0
4  enquanto tempo ≤ tempo^max faça
5      nivel ← 1
6      enquanto tempo ≤ tempo^max e nivel ≤ N faça
7          Pertubação(S, nivel)
8          S* ← BuscaLocal(S, λ, M)
9          Adiciona S* ao conjunto pareto C se for uma solução não dominada e retira as
             soluções dominadas por S*.
10         Retira solução em excesso através da métrica de crownding distance, caso tamanho
             de C ultrapasse c^max
11         se f_λ0(S*) < f_λ0(S) então
12             S ← S*
13             nivel ← 1
14         senão
15             nivel ← nivel + 1
16     λ ← número aleatório entre 0 e 1
```

O Algoritmo 3 é uma variação dos algoritmos genéticos que utiliza o critério de Dominância de Pareto para avaliar as soluções. Primeiramente se constrói uma população de indivíduos gerados aleatoriamente. Em seguida os indivíduos são ranqueados de acordo com Pareto-Dominância. Os n_{elite} melhores indivíduos são selecionados para a próxima geração. $(n_{pop} - n_{elite}) \times (1 - p_{cross})$ indivíduos são selecionados por torneio. Finalmente, $(n_{pop} - n_{elite}) \times p_{cross}$ indivíduos são gerados por cruzamento. O cruzamento é realizado utilizando o operador OX, que consiste em manter uma parte da ordem das requisições de um dos pais e preencher o resto seguindo a ordem do outro pai evitando a repetição permitindo que o filho mantenha uma permutação das requisições como solução. Para cada filho sorteado para a mutação com probabilidade, p_{mut}, é aplicado um dos movimentos propostos na Secção 5.1 selecionado de forma aleatória. Assim, a população é ranqueada novamente e é repetido o ciclo.

6. Experimentos computacionais

Nestes experimentos serão utilizadas as instâncias descritas na Seção 3. Para a implementação foi utilizada a linguagem de programação Julia (versão 1.3.0) e os experimentos foram executados em um computador com processador Intel(R) Xeon(R) E5620 2.40GHz com 115GB de memória RAM. Os testes foram executados por 15 minutos e gerados com as sementes de 0 a 5. Foram realizadas análises mono-objetivo, para comparar com a literatura atual, descritas na Seção 6.2, e análises multiobjetivo, para comprovar a capacidade do algoritmo proposto em encontrar soluções não-dominadas comparando-o com o NSGA-2, descritas na Seção 6.1.

6.1. Comparação com NSGA-2

Nesta seção o algoritmo GLS+ILS é comparado ao NSGA-2 utilizando as seguintes métricas [Riquelme et al., 2015]: (i) hipervolume, (ii) número de soluções encontradas e (iii) cobertura de conjuntos do GLS+ILS para o NSGA-2 e do NSGA-2 para o GLS+ILS.

Dado um conjunto de pontos $\mathcal{S} \in \mathbb{R}^m$ e um ponto de referência $\mathbf{r} \in \mathbb{R}^m$, o hipervolume de $\mathcal{S}$ é a medida da região dominada por $\mathcal{S}$. Dadas duas aproximações da fronteira Pareto, $\mathcal{A}$ e $\mathcal{B}$, a cobertura do conjunto $\mathcal{A}$ sobre o conjunto $\mathcal{B}$, é a proporção dos pontos de $\mathcal{B}$ dominados por pelo menos um ponto de $\mathcal{A}$.

Algoritmo 3: NSGA-2

Entrada: tempo máximo de execução $tempo_{max}$, probabilidade de cruzamento p_{cross} , probabilidade de mutação p_{mut}, tamanho da população n_{pop}, número de indivíduos elite n_{elite}

Saída: conjunto de soluções Pareto (indivíduos com rank 1 na população).

$Pop_0 \leftarrow n_{pop}$ soluções geradas aleatoriamente
Ranquear Pop_0 de acordo com a dominância de Pareto
$g = 0$
enquanto $tempo \leq tempo_{max}$ **faça**
    $Elite_{g+1} \leftarrow n_{elite}$ soluções melhor ranqueadas de Pop_g ou, caso ranks iguais, selecionados pela métrica de *crownding distance*
    $Pais_{g+1} \leftarrow$ seleção por torneio de $(1 - p_{cross}) * n_{pop}$ indivíduos da população
    $Pop_{g+1} \leftarrow Pais_{g+1} \cup Elite_{g+1}$
    $Filhos_{g+1} \leftarrow$ cruzamento por OX entre a Pop_{g+1} até que $(Filhos_{g+1} + Pop_{g+1})$ atinja n_{pop} indivíduos
    Aplicar mutação em $Filhos_{g+1}$ de acordo com p_{mut}
    $Pop_{g+1} \leftarrow Pais_{g+1} \cup Filhos_{g+1}$
    Ranquear Pop_{g+1} de acordo com a dominância de Pareto
    $g = g + 1$
return

Os resultados obtidos encontram-se na Tabela 1 e mostram que apesar do algoritmo NSGA-2 gerar em média mais soluções do que o algoritmo GLS+ILS, as soluções geradas por este último dominam, na maior parte das instâncias, as soluções do produzidas pelo NSGA-2. A análise do hipervolume também comprova que as soluções do GLS+ILS cobrem uma área maior do espaço de objetivos quando comparadas com as soluções do NSGA-2.

	Hipervolume (%)	Número de Soluções	Cobertura de Conjunto (%)
GLS+ILS	88.59	3.23	97.77
NSGA-2	38.89	8.36	23.78

Tabela 1: Resultados dos parâmetros para análise multiobjetivo.

6.2. Comparação com as Melhores Soluções Conhecidas na Literatura

Nesta Seção, avalia-se o algoritmo proposto para utilização com a formulação mono-objetivo de Heshmati et al. [2019]. Portanto, as soluções não dominadas obtidas pelos algoritmos GLS+ILS e NSGA-2 foram avaliadas de acordo com a Equação (7), com $\lambda = 0.5$, tal como definido por Heshmati et al. [2019]. O GAP entre as melhores soluções encontradas para cada instância (com $\lambda = 0.5$) e as melhores soluções conhecidas na literatura (BKS) foi calculado utilizando-se a Equação (8).

$$\text{GAP}(\mathbf{x}) = 100 \times \frac{(f_\lambda(\mathbf{x}) - BKS)}{f_\lambda(\mathbf{x})} \tag{8}$$

A Figura 7 mostra o Diagrama de Caixas das soluções obtidas nas instâncias de mesmo número de requisições[2]. Pode-se observar que, em média, as soluções obtidas pelo GLS+ILS não foram melhores que as encontradas na literatura, porém são melhores que a do NSGA-2. Por outro lado, é possível notar, que para todos os tamanhos das instâncias, em pelo menos um caso, o GLS+ILS obteve soluções melhores do que as melhores previamente conhecidas (GAP < 0). Uma vez que o GLS+ILS foi criado para problemas multiobjetivo e não para este problema mono-objetivo específico, estes resultados comprovam sua capacidade de obter soluções de alta qualidade.

[2]Resultados disponível em: `https://thomasafg@bitbucket.org/thomasafg/results_cwsp.git`

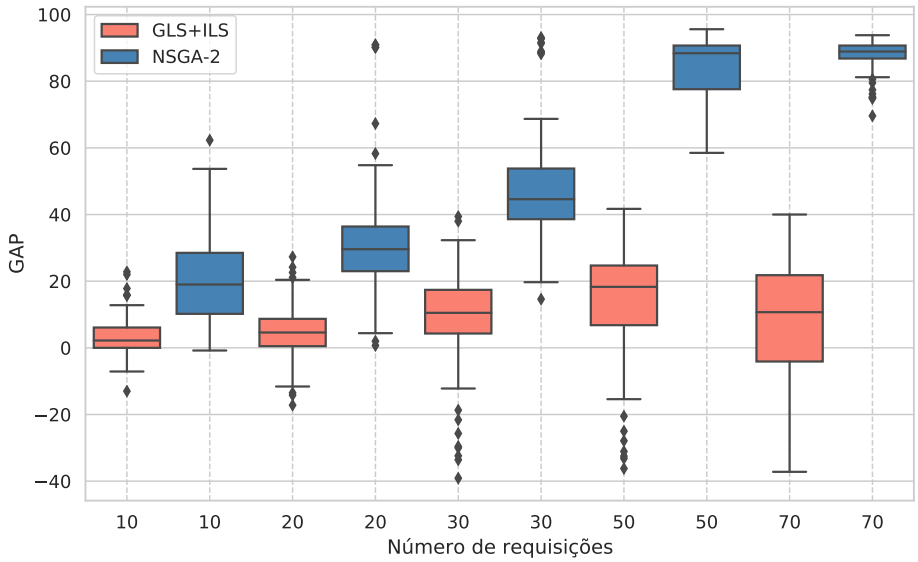


Figura 7: Diagramas de caixa formado pelos resultados obtidos para o GLS+ILS e NSGA-2.

Em geral, os experimentos mostram que a heurística GLS+ILS proposta tem vantagem sobre o NSGA2, no sentido de obter soluções mais próximas às melhores conhecidas, para o problema mono-objetivo, e suas soluções Pareto abrangerem uma área maior do espaço de soluções, no caso multiobjetivo. Esta heurística também possui vantagem sobre a heurística proposta na literatura no sentido que apresenta um conjunto de soluções não dominadas e não apenas uma única solução. Porém, analisando-as pelo tamanho das instâncias (número de requisições), em instâncias de tamanho médio e grande (número de requisições entre 30 e 70) a heurística proposta não encontra muitas soluções não dominadas, variando entre 1 e 3, sendo indesejável para um algoritmo multiobjetivo. Diferentemente das instâncias pequenas, onde este algoritmo é contundente com os algoritmos multiobjetivo.

7. Conclusão

Tendo em vista a velocidade em que a tecnologia avança nos pátios operados por pontes rolantes e o surgimento de estudos que avaliam não somente os problemas genéricos, como também as peculiaridades de cada pátio, este trabalho consegue agregar ao estudo do CWSP um caminho para a análise multiobjetivo deste problema. Foi demonstrado o conflito entre os objetivos, permitindo concluir que o CWSP é, de fato, um problema multiobjetivo. Adicionalmente, dois algoritmos multiobjetivo foram avaliados para o problema.

O algoritmo GLS+ILS, proposto neste trabalho, obteve melhores resultados do que o NSGA-2 para as instâncias do CWSP disponíveis na literatura, em termos tanto de hipervolume quanto de cobertura de conjuntos não dominados. Foi feita, ainda, uma comparação com os resultados obtidos pelo algoritmo mono-objetivo apresentado por Heshmati et al. [2019]. Apesar de abordar a versão multiobjetivo do problema, o GLS+ILS proposto foi capaz de melhorar a melhor solução conhecida da versão mono-objetivo do problema para 52 instâncias da literatura.

Para trabalhos futuros, propõe-se a implementação e análise de outras heurísticas com o objetivo de gerar um número maior de soluções Pareto ou soluções mais próximas ao conjunto Pareto-ótimo.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001. Os autores agradecem também ao Programa de Pós-Graduação em Ciência da Computação (PPGCC) da Universidade Federal de Ouro Preto pelos recursos cedidos para realização deste trabalho.

Referências

Alsheddy, A. e Tsang, E. P. K. (2010). A guided local search based algorithm for the multiobjective empowerment-based field workforce scheduling. In *2010 UK Workshop on Computational Intelligence (UKCI)*, p. 1–6.

Arroyo, J. E. C. (2002). *Heurísticas e metaheurísticas para otimização combinatória multiobjetivo*. PhD thesis, Universidade Estadual de Campinas.

Boysen, N. e Emde, S. (2016). The parallel stack loading problem to minimize blockages. *European Journal of Operational Research*, 249(2):618–627. ISSN 0377-2217.

Burke, E. K. e Bykov, Y. (2017). The late acceptance hill-climbing heuristic. *European Journal of Operational Research*, 258(1):70–78. ISSN 0377-2217.

Deb, K., Pratap, A., Agarwal, S., e Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.

Freitas, A. R. R., Fleming, P. J., e Guimarães, F. G. (2013). A non-parametric harmony-based objective reduction method for many-objective optimization. In *2013 IEEE International Conference on Systems, Man, and Cybernetics*, p. 651–656.

Heshmati, S., Toffolo, T. A. M., Vancroonenburg, W., e Vanden Berghe, G. (2019). Crane-operated warehouses: Integrating location assignment and crane scheduling. *Computers & Industrial Engineering*, 129:274–295. ISSN 0360-8352.

Ku, D. e Arthanari, T. S. (2016). On the abstraction method for the container relocation problem. *Computers & Operations Research*, 68:110–122. ISSN 0305-0548.

Li, W., Wu, Y., Petering, M., Goh, M., e Souza, R. d. (2009). Discrete time model and algorithms for container yard crane scheduling. *European Journal of Operational Research*, 198(1):165–172. ISSN 0377-2217.

Li, W., Wu, Y., Petering, M., Goh, M., Souza, R. d., e Wu, Y. (2012). A continuous time model for multiple yard crane scheduling with last minute job arrivals. *International Journal of Production Economics*, 136(2):332–343. ISSN 0925-5273.

Lourenço, H. R., Martin, O. C., e Stützle, T. (2003). *Iterated Local Search*, p. 320–353. Springer US, Boston, MA. ISBN 978-0-306-48056-0.

Riquelme, N., Von Lücken, C., e Baran, B. (2015). Performance metrics in multi-objective optimization. In *2015 Latin American Computing Conference (CLEI)*, p. 1–11.

Wu, Y., Li, W., Petering, M. E. H., Goh, M., e Souza, R. d. (2015). Scheduling multiple yard cranes with crane interference and safety distance requirement. *Transportation Science*, 49(4): 990–1005. ISSN 1526-5447.