

Um algoritmo híbrido para o problema de escalonamento integrado de satélites ágeis de observação da Terra

Yure Rocha

Universidade Federal da Paraíba, PPGI
Centro de Informática, Rua dos Escoteiros s/n, Mangabeira, 58055-000, João Pessoa, Brasil
yure.trs@gmail.com

Guilherme O. Chagas

CIRRELT e Faculté des Sciences de l'Administration
Université Laval, Québec, Canada
guilherme.oliveira-chagas.1@ulaval.ca

Leandro C. Coelho

CIRRELT, Faculté des Sciences de l'Administration e Canada Research Chair in Integrated Logistics
Université Laval, Québec, Canada
leandro.coelho@fsa.ulaval.ca

Anand Subramanian

Universidade Federal da Paraíba, Departamento de Sistemas de Computação
Centro de Informática, Rua dos Escoteiros s/n, Mangabeira, 58055-000, João Pessoa, Brasil
anand@ci.ufpb.br

RESUMO

Satélites têm aplicações importantes atualmente, como no monitoramento de desastres naturais. Este trabalho aborda o problema de escalonamento integrado de satélites ágeis de observação da Terra, cujo objetivo é maximizar a soma dos lucros associados às requisições de observação atendidas e a quantidade de dados transmitidos para as estações terrestres. Para solucionar o problema, foi proposto um algoritmo híbrido que combina um resolvedor MIP e uma heurística. Experimentos computacionais foram feitos em instâncias propostas para avaliar a performance do algoritmo em comparação com uma abordagem da literatura e também à de um resolvedor MIP. Em relação ao método da literatura, soluções de qualidade superior foram encontradas para 29 das 32 instâncias. Comparado ao resolvedor MIP, o algoritmo encontrou soluções competitivas, especialmente para as instâncias mais desafiadoras, apresentando uma melhora de até 25.93%.

PALAVRAS CHAVE. Satélites de observação da Terra. Escalonamento de satélites. Algoritmo híbrido.

Tópicos: OC – Otimização Combinatória, L&T – Logística e Transportes.

ABSTRACT

Satellites have important applications nowadays. They can be used, for example, to monitor natural disasters. This paper addresses an integrated scheduling problem of agile Earth observation satellites, whose objective is to maximize both the sum of profits associated with collected observation targets and the amount of data downloaded to ground stations. To solve the problem, we propose a hybrid algorithm that combines a MIP solver and a heuristic procedure. Computational experiments were conducted on proposed instances to evaluate the performance of the algorithm compared against an approach from the literature and with a standalone MIP solver. Regarding the literature, high-quality solutions were obtained for 29 out of 32 instances. With respect to the standalone MIP solver, the algorithm found competitive solutions, especially for the more challenging instances, with an improvement of up to 25.93%.

KEYWORDS. Earth observation satellites. Satellite scheduling. Hybrid algorithm.

Paper topics: CO – Combinatorial Optimization, L&T — Logistics and Transport.

1. Introdução

Satélites são utilizados atualmente para diversas tarefas, como por exemplo para fornecer conexão à Internet em áreas remotas ou para monitoramento da Terra. Nesse último caso, eles podem fornecer dados históricos para ajudar na prevenção de desastres naturais, como as enchentes que recentemente acometeram o estado do Rio Grande do Sul, ou no combate ao desmatamento ilegal em áreas de preservação ambiental.

Diversas empresas já lançaram constelações, ou frotas, de satélites de órbita baixa. Esses satélites são equipados com instrumentos como câmeras e têm quantidades de memória e de energia limitadas. Por um lado, a memória pode ser liberada por meio do download dos dados obtidos para estações terrestres. Por outro lado, a energia, consumida em atividades de observação, de download, e de mudança de pose do satélite, pode ser renovada periodicamente nas órbitas com iluminação solar.

Diariamente, pessoas civis ou militares enviam requisições, com diferentes prioridades, para obtenção de imagens de determinados locais do planeta. Como a órbita do satélite é conhecida, é possível saber em quais janelas de tempo um local pode ser observado, bem como quando uma operação de download pode ser realizada. Vale destacar que devido à periodicidade dos satélites, existem múltiplas janelas de tempo envolvendo o mesmo par de local (ou estação) e satélite.

O objetivo do *problema de escalonamento integrado de satélites ágeis de observação da Terra* (PEISAT), consiste em maximizar a quantidade de requisições atendidas, bem como de dados transmitidos para as estações terrestres, respeitando restrições de energia, memória e janelas de tempo. O satélite considerado neste trabalho é do tipo ágil, apresentando mobilidade nos eixos *pitch* e *roll*, conforme ilustrado na Figura 1.

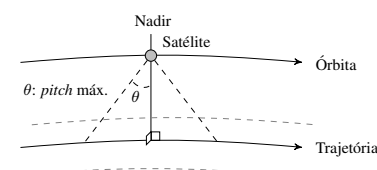


Figura 1: Ângulo de *pitch* máximo dos satélites. O ângulo de *roll* depende da direção do satélite com relação à trajetória.

A Figura 2 ilustra um exemplo de solução parcial para o PEISAT. Neste exemplo, a energia é considerada suficiente e o armazenamento interno é limitado a 15 unidades de dados. O satélite começa observando o local 1 no tempo t_1 . Depois de 7.5 unidades de tempo, ele começa a ajustar sua pose para observar o local 2, no tempo t_2 . Ao final dessa operação, o armazenamento interno do satélite (Mem) contém 15 unidades de dados, enviados para a estação g no tempo t_3 , após um tempo de transição. O alvo 3 não pode ser coletado porque o satélite precisa primeiro fazer o download dos dados associados a pelo menos um dos locais observados. O download para estação g não pode ser escalonado imediatamente no início da sua janela de tempo (J_t) devido ao longo tempo de transição necessário depois da última tarefa realizada. A solução parcial é finalizada com a aquisição do local 4, a partir do tempo t_4 . O satélite termina com 7.5 unidades de dados armazenados referentes à última observação.

Na literatura, existem diversos estudos envolvendo problemas de escalonamento de satélites, sendo possível destacar três categorias principais de problemas. A primeira classe envolve o escalonamento apenas de tarefas de observação (ver por exemplo Wolfe e Sorensen [2000]; Lemaître et al. [2002]; Liu et al. [2017]). Para uma revisão mais detalhada sobre o problema de escalonamento de satélites ágeis de observação da Terra, ver Wang et al. [2020]. O segundo *cluster* contempla problemas de escalonamento de atividades de comunicação entre satélites e estações

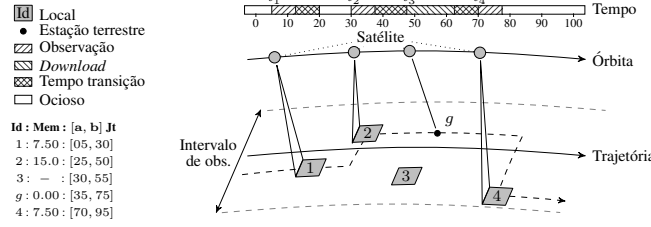


Figura 2: Exemplo de solução parcial.

terrestres, incluindo tarefas de download. Zufferey et al. [2008]; Marinelli et al. [2011]; Spangelo et al. [2015] são alguns exemplos de trabalhos nessa categoria. Os problemas de escalonamento de observações e de downloads, envolvendo satélites, são ambos \mathcal{NP} -hard [Bensanna et al., 1996; Barbulescu et al., 2004]. Já a terceira classe consiste da versão integrada de ambos os problemas. Dentre os estudos realizados, os mais próximos deste trabalho são: Wang et al. [2011]; Cho et al. [2018]; Chang et al. [2021]. Mesmo nas versões separadas dos problemas, os modelos propostos acabam intratáveis. A versão integrada é ainda mais complexa, por isso os autores muitas vezes não incorporam devidamente aspectos importantes do problema, por exemplo relacionados à energia e aos tempos de transição.

Neste trabalho, é proposto um modelo de programação linear inteira mista para resolver o problema de escalonamento integrado de satélites ágeis de observação da Terra. Para resolver o problema, utiliza-se um algoritmo híbrido, chamado *variable MIP neighborhood descent* (VMND), que combina o resolvidor MIP ao procedimento *variable neighborhood descent* (VND). Os testes foram realizados em um conjunto de instâncias propostas. Para comparar o VMND com a literatura, o modelo apresentado foi modificado para resolver o *problema de escalonamento de constelação de satélites* (PECS). A abordagem de Cho et al. [2018] para o PECS é comparada ao VMND. Foram resolvidas 32 instâncias, das quais o algoritmo híbrido apresentou soluções melhores em 29 e empatou em uma. O VMND também foi comparado ao resolvidor MIP, apresentando resultados de qualidade superior, principalmente nas instâncias de tamanho maior.

2. Formulação matemática

As seguintes premissas são adotadas neste trabalho: (i) tarefas não podem ser interrompidas depois de iniciadas; (ii) os alvos do tipo observação podem ser capturados de uma única vez; (iii) estações terrestres podem atender a no máximo um satélite por vez, assim como satélites podem executar no máximo uma operação (observação ou download) por vez; (iv); de forma análoga a Wang et al. [2011], o tempo de transição dos satélites é uma função das mudanças nos ângulos de *pitch* e de *roll*, e não há sobreposição entre janelas de tempo de observação e de download; (v) cada satélite possui um único instrumento óptico, idêntico ao dos demais; e (vi), a energia coletada pelos painéis solares dos satélites nas zonas de exposição ao sol, durante as janelas de tempo, é desconsiderada.

Para cada satélite $i \in S$ e cada alvo $\tau \in T$, $\Gamma_i^\tau = \{(a_{i\tau}^1, b_{i\tau}^1), \dots, (a_{i\tau}^{r_{i\tau}}, b_{i\tau}^{r_{i\tau}})\}$ é definido como o conjunto de janelas de tempo de oportunidades (ou o conjunto de tarefas) nas quais alvo τ é visível ao satélite i (seja para observação, se $\tau \in O$; ou para download, se $\tau \in G$). Além disso, $0 \leq a_{i\tau}^\gamma < b_{i\tau}^\gamma \leq H$ para $\gamma = 1, \dots, r_{i\tau}$ e Γ é o conjunto de todas as tarefas.

Seja o grafo ponderado, orientado e conexo $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, no qual \mathcal{V} é o conjunto de vértices e \mathcal{A} é o conjunto de arcos. Cada vértice em \mathcal{V} corresponde a uma tarefa em Γ . \mathcal{V} também contém vértices *source* e *sink*. Existe um arco entre dois vértices desse grafo caso um satélite possa executar as tarefas correspondentes em Γ , uma depois da outra. Além disso, cada vértice $p \in \mathcal{V}$

está associado à duração (δ_p) e ao lucro (ρ_p) da tarefa correspondente $\tau \in T$, e com uma janela de tempo (a_{ip}, b_{ip}) do satélite $i \in S$ que pode executar a tarefa. Considere também $\delta_0 = \delta_{n+1} = 0$, $\rho_0 = \rho_{n+1} = 0$, $(a_{i0}, b_{i0}) = (0, 0)$, $(a_{in+1}, b_{in+1}) = (H, H)$, e $\varphi_0^i = \varphi_{n+1}^i = 0, \forall i \in S$.

A Tabela 1 resume a notação matemática adotada.

Tabela 1: Notação matemática.

Dados	
c	número de alvos, $c = T $
k	número de estações terrestres, $k = G $
m	número de satélites, $m = S $
n	número de tarefas (janelas de tempo de oportunidade), $n = \Gamma $
o	número de locais de observação, $o = O $
h	número de dias do horizonte de planejamento
$[0, H]$	tempos de início e de fim do horizonte de planejamento, respectivamente
(a_{ip}, b_{ip})	janela de tempo da tarefa $p \in \mathcal{V}$, satélite $i \in S$
δ_p	duração da tarefa $p \in \mathcal{V}$, dada pela duração da tarefa relacionada $\tau \in T$
ρ_p	lucro associado à tarefa $p \in \mathcal{V}$, dado pelo lucro da tarefa relacionada $\tau \in T$
θ_{obs}	ângulo máximo de <i>pitch</i> para tarefas de observação
θ_{down}	ângulo máximo de <i>pitch</i> para tarefas de download
ϕ_{obs}	ângulo máximo de <i>roll</i> para tarefas de observação
ϕ_{down}	ângulo máximo de <i>roll</i> para tarefas de download
φ_p^i	ângulo de <i>roll</i> necessário para que o satélite $i \in S$ execute tarefa $p \in \mathcal{V}_i$
E_{obs}	consumo médio de energia por unidade de tempo para observações
E_{down}	consumo médio de energia por unidade de tempo para downloads
$\Delta\sigma$	tempo mínimo de transição das estações terrestres entre duas tarefas
E_0	energia inicial dos satélites
E_{max}	capacidade máxima de energia dos satélites
ϵ_{pq}^i	ganho de energia de $i \in S$ entre o fim da janela de tempo de p e o início da janela de tempo de q , $p, q \in \mathcal{V}_i$
L_{obs}	taxa média de ganho de dados (por segundo)
\bar{L}_{down}	taxa média de transferência de dados (por segundo)
L_0	armazenamento inicial dos satélites
L_{max}	capacidade de armazenamento dos satélites
$\omega_1, \omega_2, \omega_3, \omega_4$	constantes não-negativas (em segundos por grau), que dependem do satélite
β_1, β_2	constantes não-negativas (em segundos), que dependem do satélite
Conjuntos	
G	estações terrestres homogêneas
O	locais (ou pontos) para observação
S	satélites homogêneos
T	alvos, com $T = G \cup O$
Γ	tarefas (janelas de tempo de oportunidade)
Γ_i	tarefas visíveis ao satélite $i \in S$, $\Gamma_i \subseteq \Gamma$
Γ_i^τ	tarefas envolvendo $\tau \in T$ e satélite $i \in S$, $\Gamma_i^\tau \subseteq \Gamma_i$
\mathcal{G}	grafo, $\mathcal{G} = (\mathcal{V}, \mathcal{A})$
\mathcal{V}	vértices, $\mathcal{V} = \{0, p_1, \dots, p_n, n+1\}$
\mathcal{V}'	vértices tarefa, $\mathcal{V}' = \mathcal{V} \setminus \{0, n+1\}$
\mathcal{V}_i	vértices tarefa que podem ser executados por $i \in S$, $\{0, n+1\}$ inclusos, $\mathcal{V}_i = \mathcal{D}_i \cup \mathcal{O}_i$
\mathcal{V}'_i	vértices tarefa que podem ser executados por $i \in S$, $\mathcal{V}'_i = \mathcal{D}'_i \cup \mathcal{O}'_i$
\mathcal{A}	arcos entre vértices de \mathcal{V}
\mathcal{D}	vértices cujas tarefas correspondentes em Γ são de <i>download</i> , $\{0, n+1\}$ inclusos, $\mathcal{D} \subset \mathcal{V}$
\mathcal{D}_i	vértices tarefa de <i>download</i> do satélite $i \in S$, $\{0, n+1\}$ inclusos, $\mathcal{D}_i \subseteq \mathcal{D}$
\mathcal{D}'_i	vértices tarefa de <i>download</i> do satélite $i \in S$, $\mathcal{D}'_i = \mathcal{D}_i \setminus \{0, n+1\}$
\mathcal{D}_i^g	vértices tarefa de <i>download</i> envolvendo satélite $i \in S$ e estação $g \in G$, $\mathcal{D}_i^g \subseteq \mathcal{D}_i$
$\mathcal{D}_i^{g'}$	vértices tarefa de <i>download</i> envolvendo satélite $i \in S$ e estação $g' \in G$, $\mathcal{D}_i^{g'} = \mathcal{D}_i^g \setminus \{0, n+1\}$
\mathcal{O}	vértices cujas tarefas correspondentes em Γ são de observação, $\{0, n+1\}$ inclusos, $\mathcal{O} \subset \mathcal{V}$
\mathcal{O}_i	vértices tarefa de observação do satélite $i \in S$, $\{0, n+1\}$ inclusos, $\mathcal{O}_i \subseteq \mathcal{O}$
\mathcal{O}'_i	vértices tarefa de observação do satélite $i \in S$, $\mathcal{O}'_i = \mathcal{O}_i \setminus \{0, n+1\}$
\mathcal{O}_i^τ	vértices tarefa de observação envolvendo satélite $i \in S$ e tarefa $\tau \in T$

O PEISAT é modelado como um problema de programação não linear inteira mista (1)–(28). Nessa formulação, as seguintes variáveis são utilizadas: (i) x_{pq}^i – variável de decisão binária

usada para indicar se tarefas $p, q \in \mathcal{V}_i$ são adotadas uma depois da outra pelo satélite $i \in S$. x_{pq}^i é criada somente se a tarefa q pode ser realizada depois da tarefa p ; (ii) y_{pq}^i – variável de decisão binária utilizada para indicar se tarefas p e q ($p, q \in \mathcal{O}_i$ ou $p, q \in \mathcal{D}_i$) são escolhidas como tarefas consecutivas pelo satélite $i \in S$. Análogo ao x_{pq}^i , y_{pq}^i é criada somente se a tarefa q pode ser executada após a tarefa p . Ademais, y_{pq}^i é adicionada se restrições de tempo de transição são necessárias entre as tarefas p e q ; e (iii) $z_{pq}^{gii'}$ – variável de decisão binária usada para indicar se tarefas de download $p \in \mathcal{D}_i^g$ e $q \in \mathcal{D}_{i'}^g$, envolvendo a estação $g \in G$, são adotadas pelos satélites i e i' , respectivamente. A variável $z_{pq}^{gii'}$ existe apenas se restrições de tempo de transição são necessárias para essas tarefas

As variáveis de decisão contínuas usadas no modelo (1)–(28) são descritas a seguir: (i) d_p^i – variável contínua e não negativa para a duração da tarefa $p \in \mathcal{V}_i$ do satélite $i \in S$. Se $p \in \mathcal{O}_i'$, $d_p^i = \delta_p$; caso contrário, se $p \in \mathcal{D}_i'$, $\delta_p \leq d_p^i \leq \frac{L_{max}}{L_{down}}$. Ademais, se $p \in \{0, n+1\}$, $d_p^i = 0$; (ii) l_p^i – variável contínua e não negativa que indica a quantidade de memória utilizada pelo satélite $i \in S$ no final da execução da tarefa $p \in \mathcal{V}_i$; (iii) e_p^i – variável contínua e não negativa que indica a quantidade de energia do satélite $i \in S$ no final da execução da tarefa $p \in \mathcal{V}_i$; (iv) t_p^i – variável contínua e não negativa utilizada para indicar o tempo de início da tarefa $p \in \mathcal{V}_i'$, dentro da janela de tempo (a_{ip}, b_{ip}) , pelo satélite $i \in S$. Se $p \in \{0, n+1\}$, $a_{ip} \leq t_p^i \leq b_{ip}$; (v) ϑ_p^i – variável contínua e não negativa representando o ângulo de *pitch* do satélite $i \in S$ para executar a tarefa $p \in \mathcal{V}_i$; (vi) Δl_p^i – variável contínua e não negativa indicando a alteração no armazenamento do satélite $i \in S$ após execução da tarefa $p \in \mathcal{V}_i$; (vii) Δe_p^i – variável contínua e não negativa indicando a quantidade de energia consumida pelo satélite $i \in S$ na tarefa $p \in \mathcal{V}_i$; e (viii) Δs_{pq}^i – variável contínua e não negativa para o tempo de transição necessário pelo satélite $i \in S$ entre tarefas $p, q \in \mathcal{V}_i$.

As variáveis x_{pq}^i e y_{pq}^i , bem como as restrições associadas a elas, consideram $p \neq q$. De forma similar, as variáveis $z_{pq}^{gii'}$, e todas as restrições associadas a elas, assumem $i \neq i'$. A formulação não linear do problema é apresentada a seguir.

$$\max \sum_{i \in S} \sum_{p \in \mathcal{O}_i \setminus \{n+1\}} \sum_{q \in \mathcal{V}_i \setminus \{0\}} (\rho_p + \delta_p) x_{pq}^i - \frac{1}{L_{down}} \sum_{i \in S} l_{n+1}^i \quad (1)$$

subject to

$$\sum_{p \in \mathcal{V}_i \setminus \{n+1\}} x_{pq}^i - \sum_{r \in \mathcal{V}_i \setminus \{0\}} x_{qr}^i = 0, \quad i \in S, q \in \mathcal{V}_i' \quad (2)$$

$$\sum_{q \in \mathcal{V}_i \setminus \{0\}} x_{0q}^i = 1, \quad i \in S \quad (3)$$

$$\sum_{p \in \mathcal{V}_i \setminus \{n+1\}} x_{pn+1}^i = 1, \quad i \in S \quad (4)$$

$$\sum_{i \in S} \sum_{p \in \mathcal{O}_i'} \sum_{q \in \mathcal{V}_i \setminus \{0\}} x_{pq}^i \leq 1, \quad \tau \in T \quad (5)$$

$$y_{pq}^i + y_{qp}^i = \min \left\{ \sum_{r \in \mathcal{V}_i \setminus \{0\}} x_{pr}^i, \sum_{u \in \mathcal{V}_i \setminus \{0\}} x_{qu}^i \right\}, \quad i \in S, p \in \mathcal{V}_i \setminus \{n+1\}, q \in \mathcal{V}_i' \quad (6)$$

$$\vartheta_p^i = \theta_{obs} \left\{ 2 \frac{t_p^i - a_{ip}}{b_{ip} - a_{ip}} - 1 \right\}, \quad i \in S, p \in \mathcal{O}_i \quad (7)$$

$$\vartheta_p^i = \theta_{down} \left\{ 2 \frac{t_p^i - a_{ip}}{b_{ip} - a_{ip}} - 1 \right\}, \quad i \in S, p \in \mathcal{D}_i \quad (8)$$

$$\Delta s_{pq}^i = \omega_1 |\varphi_p^i - \varphi_q^i| + \omega_2 |\vartheta_p^i - \vartheta_q^i| + \beta_1, \quad i \in S, p, q \in \mathcal{O}_i \quad (9)$$

$$\Delta s_{pq}^i = \omega_3 |\varphi_p^i - \varphi_q^i| + \omega_4 |\vartheta_p^i - \vartheta_q^i| + \beta_2, \quad i \in S, p, q \in \mathcal{D}_i \quad (10)$$

$$y_{pq}^i (t_p^i + d_p^i + \Delta s_{pq}^i - t_q^i) \leq 0, \quad i \in S, p \in \mathcal{V}_i \setminus \{n+1\}, q \in \mathcal{V}_i \setminus \{0\} \quad (11)$$

$$z_{pq}^{gii'} + z_{qp}^{gi'i} = \min \left\{ \sum_{r \in \mathcal{V}_i \setminus \{0\}} x_{pr}^i, \sum_{u \in \mathcal{V}_{i'} \setminus \{0\}} x_{qu}^{i'} \right\}, \quad g \in G, i, i' \in S, p \in \mathcal{D}_i^g, q \in \mathcal{D}_{i'}^g \quad (12)$$

$$z_{pq}^{gii'} (t_p^i + d_p^i + \Delta\sigma - t_q^{i'}) \leq 0, \quad g \in G, i, i' \in S, p \in \mathcal{D}_i^g, q \in \mathcal{D}_{i'}^g \quad (13)$$

$$a_{ip} \leq t_p^i \leq b_{ip} - \delta_p, \quad i \in S, p \in \mathcal{V}_i' \quad (14)$$

$$t_p^i + d_p^i \leq b_{ip}, \quad i \in S, p \in \mathcal{D}_i' \quad (15)$$

$$\delta_p \leq d_p^i \leq \frac{L_{max}}{L_{down}}, \quad i \in S, p \in \mathcal{D}_i' \quad (16)$$

$$l_0^i = L_0, \quad i \in S \quad (17)$$

$$\delta_p \leq l_p^i \leq L_{max}, \quad i \in S, p \in \mathcal{O}_i' \quad (18)$$

$$0 \leq l_p^i \leq L_{max}, \quad i \in S, p \in \mathcal{D}_i \quad (19)$$

$$\Delta l_p^i = \begin{cases} \bar{L}_{obs} d_p^i, & \text{se } p \in \mathcal{O}_i, \\ -\bar{L}_{down} d_p^i, & \text{caso contrário.} \end{cases} \quad i \in S, p \in \mathcal{V}_i \setminus \{0\} \quad (20)$$

$$x_{pq}^i (l_p^i + \Delta l_q^i - l_q^i) = 0, \quad i \in S, p \in \mathcal{V}_i \setminus \{n+1\}, q \in \mathcal{V}_i \setminus \{0\} \quad (21)$$

$$e_0^i = E_0, \quad i \in S \quad (22)$$

$$0 \leq e_p^i \leq E_{max}, \quad i \in S, p \in \mathcal{V}_i \quad (23)$$

$$\Delta e_p^i = \begin{cases} \bar{E}_{obs} d_p^i, & \text{se } p \in \mathcal{O}_i, \\ \bar{E}_{down} d_p^i, & \text{caso contrário.} \end{cases} \quad i \in S, p \in \mathcal{V}_i \setminus \{0\} \quad (24)$$

$$x_{pq}^i (e_q^i - \min\{E_{max}^i, e_p^i + \epsilon_{pq}^i - \Delta e_q^i\}) = 0, \quad i \in S, p \in \mathcal{V}_i \setminus \{n+1\}, q \in \mathcal{V}_i \setminus \{0\} \quad (25)$$

$$x_{pq}^i (t_p^i + d_p^i - t_q^i) \leq 0, \quad i \in S, p \in \mathcal{V}_i \setminus \{n+1\}, q \in \mathcal{V}_i \setminus \{0\} \quad (26)$$

$$\Delta l_p^i, v_p^i \in \mathbb{R}, d_p^i, e_p^i, l_p^i, t_p^i, \Delta e_p^i \in \mathbb{R}_{\geq 0}, \quad \forall i \in S, \forall p \in \mathcal{V}_i \quad (27)$$

$$x_{pq}^i \in \{0, 1\}, y_{pq}^i \in \{0, 1\}, \Delta s_{pq}^i \in \mathbb{R}_{\geq 0}, \quad \forall i \in S, \forall p, q \in \mathcal{V}_i \quad (28)$$

$$z_{pq}^{gii'} \in \{0, 1\}, \quad \forall g \in G, \forall i, i' \in S, \forall p \in \mathcal{D}_i^g, \forall q \in \mathcal{D}_{i'}^g. \quad (29)$$

A função objetivo (1) maximiza a soma dos lucros dos locais observados e a quantidade de dados transmitidos para as estações terrestres. As restrições de conservação de fluxo (2) garantem que a quantidade de fluxo entrando e saindo de um nó sejam iguais, exceto pelo nó de *source* (3) e o nó de *sink* (4). As restrições (5) definem a quantidade máxima de observações por local. Diferentemente das variáveis x_{pq}^i , que consideram apenas tarefas adotadas uma após a outra, as variáveis y_{pq}^i levam em conta quaisquer p e q . Isso é modelado por meio das restrições (6). Os ângulos de *pitch* do satélite $i \in S$ no início de tarefas de observação e de download são definidos pelas restrições (7) e (8), respectivamente. O tempos de transição Δs_{pq}^i do satélite $i \in S$ entre tarefas de observação ($p, q \in \mathcal{O}_i$) e de download ($p, q \in \mathcal{D}_i$) são dados pelas restrições (9) e (10), respectivamente. Esses tempos Δs_{pq}^i são então utilizados nas restrições (11) para garantir que o satélite consiga ajustar sua pose entre tarefas de observação ou de download.

As restrições (12) modelam variáveis de decisão $z_{pq}^{gii'}$, usadas nas restrições (13) para assegurar que o tempo de transição da estação g , entre duas atividades de download, seja respeitado. As desigualdades (14) asseguram que o tempo de início de qualquer tarefa p esteja dentro da janela de tempo correspondente. Os limites envolvendo variáveis d_p^i são estabelecidos pelas restrições (15) e (16). A quantidade de memória ocupada inicialmente pelos satélites é estabelecida nas restrições (17), enquanto restrições (18) e (19) impõem limites mínimos e máximos na capacidade de armazenamento dos satélites. De maneira análoga, as restrições (22) definem a quantidade inicial de energia dos satélites e as restrições (23) configuram as quantidades mínimas e máximas de energia armazenada na bateria. As variações de armazenamento e de energia são tratadas pelas restrições (21) e (25), respectivamente. As variáveis x_{pq}^i , t_p^i e t_q^i são conectadas por meio das restrições (26). Por último, as restrições (27)–(29) definem os domínios das variáveis.

Todas as restrições não-lineares apresentadas podem ser linearizadas. Em particular, as restrições (11), (13), (21), (25) e (26) podem ser linearizadas com restrições *Miller–Tucker–Zemlin*.

Dessa forma, o modelo final pode ser representado por um problema de programação linear inteira mista. No restante deste trabalho, a versão linear do modelo será considerada.

3. Algoritmo proposto

Uma estratégia eficiente com tarefas fictícias foi implementada com o intuito de reduzir a quantidade de variáveis e de restrições, mas preservando o espaço de soluções original associado a uma instância. Tal estratégia consiste em criar um novo grafo no qual, exceto pelos vértices imediatamente após o *source* e logo antes do *sink*, um vértice fictício (ou *dummy*) está associado a cada vértice, conforme mostra a Figura 3.

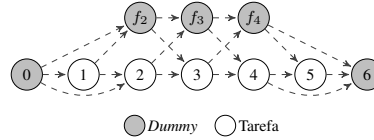


Figura 3: Estratégia eficiente para uma instância com 5 tarefas.

Na Figura 3, os nós f_1, f_2 e f_3 correspondem a nós *dummy* e os arcos indicam quais variáveis x_{pq}^i são criadas. A explicação de alguns pormenores dessa estratégia, bem como a prova de que ela minimiza a quantidade de arcos criados, foge do escopo deste trabalho.

3.1. A heurística VMND

A ideia principal da heurística VMND utilizada neste trabalho (Algoritmo 1) é explorar, de maneira controlada, o espaço de soluções associado a uma instância. Isso é feito por meio das estruturas de vizinhança, que fixam os *bounds* das variáveis x_{pq}^i antes de execuções do resolvidor MIP.

Algorithm 1 VMND

```

1: procedure VMND( $\mathcal{N}, time\_limit$ )
2:    $time\_limit' \leftarrow$  Divide  $time\_limit$  proporcionalmente entre  $build\_initial\_solution$  e as vizinhanças.
3:    $runtime \leftarrow build\_initial\_solution(time\_limit'^{(0)})$ 
4:    $rem\_time \leftarrow time\_limit - runtime$ 
5:   for  $k = 1$  to  $|\mathcal{N}|$ 
6:      $time\_limit' \leftarrow$  Divide  $rem\_time$  proporcionalmente entre as vizinhanças restantes.
7:      $s^*, runtime \leftarrow \mathcal{N}^{(k)}(time\_limit'^{(k)})$ 
8:      $rem\_time \leftarrow rem\_time - runtime$ 
9:   end
10:  return  $s^*$ 
11: end

```

O procedimento VMND começa dividindo o tempo limite (linha 2) entre (i) a função que constrói a solução inicial e (ii) as estruturas de vizinhança. O critério dessa divisão será discutido na Seção 4.2. Em seguida, uma solução inicial é construída (linha 3) em no máximo $time_limit'^{(0)}$. O laço de repetição principal do algoritmo (linhas 5–9) é executado $|\mathcal{N}|$ vezes. Em cada iteração, o tempo limite que resta é proporcionalmente dividido entre as vizinhanças remanescentes (linha 6). A vizinhança selecionada é aplicada (linha 7) e o tempo restante calculado novamente (linha 8). No final do algoritmo, a solução incumbente é retornada (linha 10).

3.2. Estruturas de vizinhança

As estruturas de vizinhança controlam o esforço computacional da execução do resolvidor MIP alterando os *bounds* das variáveis x_{pq}^i . Há duas possibilidades para esses limites: (i) livres – limites inferior e superior iguais a 0 e 1, respectivamente; e (ii) fixos – ambos os limites inferior e superior iguais a 0 ou a 1, dependendo do valor encontrado pelo resolvidor para x_{pq}^i na execução

anterior. No começo de cada iteração do VMND, os limites de todas as variáveis x_{pq}^i estão fixos, de acordo com a solução incumbente, e são liberados dependendo da estrutura de vizinhança a ser aplicada.

- $\alpha_{\kappa_unsch_targets}$: os limites são liberados para as variáveis ou arcos x_{pq}^i envolvendo tarefas da solução incumbente e (i) tarefas do tipo download ou *dummy*, ou (ii) tarefas associadas com $\alpha\%$ dos alvos não escalonados, ordenados em ordem não-decrescente. Não é possível calcular, de maneira direta, a porcentagem relativa dos *bounds* liberados por iteração do laço principal desta vizinhança.
- *intra_sat*: libera os *bounds* dos arcos associados a um satélite por iteração. Este procedimento tem m iterações no seu laço principal. Em cada uma das iterações, os limites de $\frac{1}{m}\%$ do número total de arcos são liberados.
- λ_mult_sats : divide as tarefas disponíveis para cada satélite em λ intervalos. Em cada iteração, os limites são liberados para as variáveis x_{pq}^i cujas tarefas p e q estejam ambas no intervalo selecionado. Desta forma, a estrutura λ_mult_sats tem λ iterações e em cada uma, os limites liberados representam $\frac{1}{\lambda m}\%$ do total de tarefas.
- *doub_sats*: similar à *intra_sat*, mas considerando, em vez de um único satélite, todas as combinações de pares de satélites. O procedimento leva $\binom{m}{2} = \frac{m!}{2!(m-2)!}$ iterações e, em cada uma, os *bounds* associados a $\frac{2}{m}\%$ de todas as tarefas são liberados.

O esforço computacional aumenta exponencialmente com a quantidade de *bounds* liberados. Como o intuito do VMND é fazer pequenas melhorias em cada iteração, também são consideradas algumas variações dos operadores $\alpha_{\kappa_unsch_targets}$, *intra_sat* e *doub_sats*, que liberam os limites de menos arcos: (i) *dwn_tasks* – um caso especial da vizinhança $\alpha_{\kappa_unsch_targets}$, na qual $\alpha = 1$, $\kappa = 1$, e somente os limites de arcos na solução ou entre tarefas na solução e tarefas de download são liberados; (ii) μ_intra_sat – similar a *intra_sat*, mas dividindo as tarefas disponíveis para cada satélite em dois intervalos (μm iterações) com os mesmos números de tarefas, cada. Os *bounds* são liberados para os arcos em um intervalo ($\frac{1}{\mu m}\%$) por vez; e (iii) π_doub_sats – difere de *doub_sats* porque os arcos associados a cada satélite são divididos em π intervalos ($2\binom{m}{\pi}$ iterações). Desse modo, os limites de $\frac{2\pi}{m}\%$ dos arcos são liberados por iteração.

Nas vizinhanças baseadas em intervalos (i.e., μ_intra_sat , λ_mult_sats e π_doub_sats), os limites de um arco não são liberados se alguma das tarefas é do tipo observação e o local associado já foi escalonado por um arco fora do intervalo selecionado.

4. Experimentos computacionais

O algoritmo proposto foi implementado com a linguagem de programação C++ e compilado com o g++, versão 12.3.0. Além disso, foi utilizado o Gurobi, versão 11.0.1, para resolver os problemas de programação linear. Todos os testes descritos nesta seção, bem como todos os algoritmos, foram executados em uma mesma máquina Intel® Core™ i9-13900K CPU 3.0 GHz, com 32 *threads* e 128 GB de memória RAM.

Para fins de comparação com a literatura, é possível adaptar o modelo discutido para resolver um outro problema chamado *problema de escalonamento de constelação de satélites* (PECS). Cho et al. [2018] utilizaram uma abordagem de duas fases (chamada *programação linear binária em duas etapas* – PLBDE) para resolver o PECS, na qual a solução para o problema de escalonamento de atividades de download é utilizada como entrada para o problema de escalonamento de observações. Os autores também implementaram um algoritmo *first-in first-out* (FIFO). A única

diferença entre o PEISAT e o PECS é que este último considera satélites não-ágeis. Sendo assim, apenas restrições (9) e (10) precisam ser modificadas.

O modelo proposto (1)–(28), apesar de tratar diversos aspectos relevantes do problema real, exige muito esforço computacional. É possível reduzir parte deste esforço, eliminando-se as variáveis y_{pq}^i . Tendo por base a desigualdade triangular, as variáveis y_{pq}^i e as restrições (6) e (11) são removidas do modelo e as seguintes restrições são adicionadas.

$$\Delta s_{pq}^i = \max\{\omega_1, \omega_3\}|\varphi_p^i - \varphi_q^i| + \max\{\omega_2, \omega_4\}|\vartheta_p^i - \vartheta_q^i| + \max\{\beta_1, \beta_2\}/2, \quad i \in S, p \in O_i, q \in D_i \quad (30)$$

$$\Delta s_{pq}^i = \max\{\omega_1, \omega_3\}|\varphi_p^i - \varphi_q^i| + \max\{\omega_2, \omega_4\}|\vartheta_p^i - \vartheta_q^i| + \max\{\beta_1, \beta_2\}/2, \quad i \in S, p \in D_i, q \in O_i \quad (31)$$

$$x_{pq}^i(t_p^i + d_p^i + \Delta s_{pq}^i - t_q^i) \leq 0, \quad i \in S, p \in \mathcal{V}_i \setminus \{n+1\}, q \in \mathcal{V}_i \setminus \{0\} \quad (32)$$

4.1. Benchmark

Para criar as instâncias para ambos os problemas PEISAT e PECS, foi considerado o conjunto de dados utilizado em Cho et al. [2018], projetado a partir de missões do mundo real. Mais especificamente, 32 instâncias foram criadas, divergindo em termos da quantidade de locais para observação o , de satélites m , de estações terrestres k e de dias h no horizonte de planejamento. Além disso, as seguintes modificações foram realizadas: (i) números associados às janelas de tempo arredondados para o inteiro mais próximo; e (ii) tarefas de observação com *overlaps* com tarefas de download removidas. Informações adicionais sobre as instâncias podem ser encontradas em Cho et al. [2018].

4.2. Calibração dos parâmetros

As instâncias do *benchmark* foram divididas em 8 grupos, de acordo com o número de locais para observação e de satélites. Para a calibração dos parâmetros, a instância com o maior horizonte de planejamento e menor número de estações terrestres foi selecionada de cada grupo. Como o objetivo do VMND é relacionado à qualidade das soluções, o resolvidor foi configurado para encontrar, rapidamente, boas soluções para o problema.

Caso o resolvidor não consiga encontrar uma solução inicial em $time_limit^{(0)}$, uma solução vazia é adotada, apenas com as tarefas de *source* e de *sink* conectadas sem nenhuma tarefa intermediária de observação ou de download.

Com exceção de algumas vizinhanças, cujos esforços computacionais não podem ser diretamente deduzidos, os tempos limites das iterações dos movimentos de vizinhança são configurados de forma proporcional à quantidade de *bounds* liberados pelas estruturas. O tempo limite para uma iteração de um movimento que libera os limites dos arcos associados a um único satélite é chamado de t_{sat} . Esse valor é recalculado a cada iteração do laço principal do VMND. Os seguintes limites de tempo foram adotados para cada iteração das estruturas: $tl = t_{sat}$ para *build_initial_solution*, $\alpha_k_unsch_targets$, *intra_sat* e *dwn_tasks*; $tl = \frac{mt_{sat}}{\lambda}$ para λ_mult_sats ; $tl = 2t_{sat}$ para *doub_sats*; $tl = \frac{t_{sat}}{\mu}$ para μ_intra_sat ; e $tl = \frac{2t_{sat}}{\pi}$ para π_doub_sats .

Foi usada uma abordagem incremental, similar à descrita em Kramer e Subramanian [2019], para selecionar a ordem de aplicação das estruturas de vizinhança. Os operadores foram aplicados em ordem não-decrescente de esforço computacional (determinado pela porcentagem de arcos cujos limites foram liberados). Há algumas exceções para essa regra, como por exemplo as vizinhanças $\alpha_k_unsch_targets$ e *dwn_tasks*, nas quais o esforço computacional depende da solução incumbente.

A Tabela 2 apresenta os resultados obtidos para diferentes configurações de estruturas de vizinhanças do VMND aplicado ao PECS. O “Tempo limite (s)” leva em conta o tempo total de

Tabela 2: Diferentes configurações de estruturas de vizinhança do VMND para o PECS.

Configuração	Vizinhanças (\mathcal{N})	Tempo limite (s)	Gap (%)
1	70_2_unsch_targets + 80_2_unsch_targets + 90_2_unsch_targets + 100_2_unsch_targets	600	2.43
2	60_2_unsch_targets +	600	1.99
3	50_2_unsch_targets +	600	2.11
4	60_2_unsch_targets + 70_2_unsch_targets + 80_2_unsch_targets + 90_2_unsch_targets + 100_2_unsch_targets	3600	1.84
5	+ dwn_tasks	3600	0.58
6	+ 2_intra_sat	3600	0.26
7	+ intra_sat	3600	0.25
8	+ dwn_tasks	3600	-0.09
9	+ 4_mult_sats	3600	-0.34
10	+ dwn_tasks	3600	-0.34
11	+ 2_doub_sats + doub_sats	3600	-0.65
12	+ dwn_tasks	3600	-0.68
13	+ 2_mult_sats	3600	-0.82
14	+ 1_mult_sats	3600	-0.89
15	+ dwn_tasks	3600	-0.83

execução do algoritmo, enquanto o “Gap (%)” indica a média dos gaps percentuais entre o valor encontrado pelo VMND e a melhor solução entre a heurística FIFO e o PLBDE.

A busca local é iniciada com a aplicação da estrutura $\alpha_{\kappa_unsch_targets}$ à solução inicial. Para simplificar, o nome α_{κ} será utilizado em vez de $\alpha_{\kappa_unsch_targets}$. As configurações 1, 2 e 3 da Tabela 2 contêm os resultados de diferentes combinações de α_{κ} , com tempo limite total de 600 segundos. O valor $\kappa = 2$ foi adotado por oferecer transações mais suaves entre execuções consecutivas da vizinhança α_{κ} . Inicialmente, na configuração 1, as vizinhanças 70_2, 80_2, 90_2 e 100_2 são aplicadas, nessa ordem. Em seguida, na configuração 2, a vizinhança 60_2 é adicionada antes das vizinhanças da configuração 1. De forma análoga, a vizinhança 50_2 é inserida, na configuração 3, antes das vizinhanças da configuração 2. Como há piora no “Gap (%)”, a configuração 3 é descartada (riscado na vizinhança 50_2 adicionada na configuração 3). A configuração 4 contém o resultado do VMND com configuração 2 e tempo limite total de 3600 segundos. Na configuração 5, a vizinhança dwn_tasks é inserida após as vizinhanças da configuração 4. De agora em diante, dwn_tasks será testada após a exploração de vizinhanças semelhantes (configurações 8, 10, 12 e 15) para facilitar o download de dados, o que corresponde a metade do objetivo do problema. O processo de adição de vizinhança após as vizinhanças da configuração anterior, com descarte quando há piora, continua até a configuração 15, com a melhor sequência ocorrendo na configuração 14.

4.3. Comparação com a literatura

A Tabela 3 apresenta os resultados da heurística VMND aplicada ao PECS. A coluna “Obj.” reporta o valor da função objetivo obtido pelo VMND, enquanto o “Gap (%)” foi calculado de forma análoga ao da Seção 4.2.

Como retratado na Tabela 3, a heurística alcançou os melhores resultados para quase todas as instâncias. Em apenas 3 instâncias o VMND não foi capaz de obter soluções melhores, empatando em uma delas. Com relação às instâncias com 200 e 400 locais para observação, a heurística VMND obteve uma média de gaps -1.97% melhor com relação aos algoritmos da literatura. Já nas instâncias com 600 e 800 locais para observação, a média dos gaps foi de -0.44% .

4.4. Resultados

A Tabela 4 mostra os resultados do algoritmo VMND comparado ao MIP *solver*, ambos resolvendo o PEISAT.

O VMND apresentou resultados de qualidade superior na maioria das instâncias, principalmente naquelas com 600 e 800 locais para observação, nas quais a média dos gaps foi de

Tabela 3: Resultados obtidos pelo VMND para o PECS.

Instância				FIFO	PLBDE	VMND		FIFO	PLBDE	VMND	
<i>c</i>	<i>m</i>	<i>k</i>	<i>h</i>			Obj.	Gap(%)			Obj.	Gap(%)
200	1	1	1	1440	1600	1640	−2.50	600	3	2	2
200	1	1	2	3290	3730	3820	−2.41	600	3	2	3
200	1	2	1	1970	2120	2190	−3.30	600	3	4	2
200	1	2	2	3910	4150	4290	−3.37	600	3	4	3
200	3	1	1	4760	5192	5360	−3.24	600	6	2	2
200	3	1	2	9050	9910	10030	−1.21	600	6	2	3
200	3	2	1	6000	6330	6540	−3.32	600	6	4	2
200	3	2	2	9870	10480	10620	−1.34	600	6	4	3
400	1	1	1	1590	1850	1850	0.00	800	3	2	2
400	1	1	2	3650	4440	4460	−0.45	800	3	2	3
400	1	2	1	2370	2580	2630	−1.94	800	3	4	2
400	1	2	2	4820	5370	5490	−2.23	800	3	4	3
400	3	1	1	5340	6012	6180	−2.79	800	6	2	2
400	3	1	2	11140	12840	13010	−1.32	800	6	2	3
400	3	2	1	7140	7860	7960	−1.27	800	6	4	2
400	3	2	2	12790	14490	14610	−0.83	800	6	4	3
Média				5571	6185	6292	−1.97	Média			
								25568	29336	29460	−0.44

Tabela 4: Resultados obtidos pelo VMND para o PEISAT.

Instância				MIP solver		VMND		Instância				MIP solver		VMND	
<i>c</i>	<i>m</i>	<i>k</i>	<i>h</i>	Obj.	Gap _{opt} (%)	Obj.	Gap(%)	<i>c</i>	<i>m</i>	<i>k</i>	<i>h</i>	Obj.	Gap _{opt} (%)	Obj.	Gap(%)
200	1	1	1	1540	41.35	1540	0.00	600	3	2	2	14130	47.05	14280	−1.06
200	1	1	2	3420	32.33	3420	0.00	600	3	2	3	19010	37.59	19300	−1.53
200	1	2	1	1940	15.07	1940	0.00	600	3	4	2	16770	24.14	16910	−0.83
200	1	2	2	3810	18.83	3810	0.00	600	3	4	3	22110	18.30	22370	−1.18
200	3	1	1	4880	27.10	4880	0.00	600	6	2	2	26860	22.32	27570	−2.64
200	3	1	2	9070	10.60	9070	0.00	600	6	2	3	26420	29.98	33270	−25.93
200	3	2	1	5790	12.67	5790	0.00	600	6	4	2	29460	11.18	30270	−2.75
200	3	2	2	9660	3.80	9660	0.00	600	6	4	3	33910	3.37	34030	−0.35
400	1	1	1	1770	81.11	1770	0.00	800	3	2	2	15380	59.12	15790	−2.67
400	1	1	2	4110	63.04	4110	0.00	800	3	2	3	20850	52.10	21540	−3.31
400	1	2	1	2260	44.14	2260	0.00	800	3	4	2	18270	34.42	18610	−1.86
400	1	2	2	4730	40.36	4730	0.00	800	3	4	3	24210	31.16	24970	−3.14
400	3	1	1	5790	69.90	5790	0.00	800	6	2	2	26680	56.25	31190	−16.90
400	3	1	2	11900	38.30	11940	−0.34	800	6	2	3	30890	49.10	36050	−16.70
400	3	2	1	6980	40.84	6980	0.00	800	6	4	2	32150	29.34	35910	−11.70
400	3	2	2	12930	27.84	12990	−0.46	800	6	4	3	40580	13.39	42400	−4.48
Média				5661	34.45	27.37	−0.05	Média				24855	32.42	26529	−6.06

−6.06%, com o menor valor do gap chegando a −25.93% em uma das instâncias. Nas instâncias com 200 locais, o VMND e o resolvidor MIP empataram. Já nas instâncias com 400 pontos para observação, o VMND se mostrou superior em duas instâncias, com gaps de −0.34% e de −0.46%.

5. Conclusão e trabalhos futuros

Este trabalho investigou a aplicação de um algoritmo híbrido, que combina um resolvidor MIP à heurística *variable neighborhood descent* (VND), aplicada ao *problema de escalonamento integrado de satélites de observação da Terra* (PEISAT). O PEISAT é modelado como um problema de programação linear inteira mista. Para realizar os experimentos computacionais, instâncias foram criadas a partir de um conjunto de dados relacionados a um trabalho da literatura. Também foi discutido como adaptar o modelo proposto para resolver o *problema de escalonamento de constelação de satélites* (PECS). Por último, os resultados obtidos pelo VMND foram comparados aos resultados dos métodos propostos por Cho et al. [2018] no PECS. Além disso, o VMND também foi confrontado com o resolvidor MIP. Em ambos os casos, o algoritmo híbrido apresentou resultados melhores.

Trabalhos futuros incluem o estudo de alvos do tipo “área”, que não podem ser observados de uma única vez. Também é relevante estudar como incluir nas restrições a energia consumida pelos satélites nas mudanças de pose entre atividades consecutivas.

Referências

- Barbulescu, L., Watson, J.-P., Whitley, L. D., e Howe, A. E. (2004). Scheduling space-ground communications for the air force satellite control network. *Journal of Scheduling*, 7:7–34.
- Bensanna, E., Verfaillie, G., Agnès, J., Bataille, N., e Blumstein, D. (1996). Exact and approximate methods for the daily management of an earth observation satellite. In *Proc. of the 4 th Int. Symposium on Space Mission Operations and Ground Data Systems, Munich, Germany*.
- Chang, Z., Zhou, Z., Xing, L., e Yao, F. (2021). Integrated scheduling problem for earth observation satellites based on three modeling frameworks: an adaptive bi-objective memetic algorithm. *Memetic Computing*, 13(2):203–226.
- Cho, D.-H., Kim, J.-H., Choi, H.-L., e Ahn, J. (2018). Optimization-based scheduling method for agile earth-observing satellite constellation. *Journal of Aerospace Information Systems*, 15(11):611–626.
- Kramer, A. e Subramanian, A. (2019). A unified heuristic and an annotated bibliography for a large class of earliness-tardiness scheduling problems. *Journal of Scheduling*, 22(1):21–57.
- Lemaître, M., Verfaillie, G., Jouhaud, F., Lachiver, J.-M., e Bataille, N. (2002). Selecting and scheduling observations of agile satellites. *Aerospace Science and Technology*, 6(5):367–381.
- Liu, X., Laporte, G., Chen, Y., e He, R. (2017). An adaptive large neighborhood search metaheuristic for agile satellite scheduling with time-dependent transition time. *Computers & Operations Research*, 86:41–53.
- Marinelli, F., Nocella, S., Rossi, F., e Smriglio, S. (2011). A lagrangian heuristic for satellite range scheduling with resource constraints. *Computers & Operations Research*, 38(11):1572–1583.
- Spangelo, S., Cutler, J., Gilson, K., e Cohn, A. (2015). Optimization-based scheduling for the single-satellite, multi-ground station communication problem. *Computers & Operations Research*, 57:1–16.
- Wang, P., Reinelt, G., Gao, P., e Tan, Y. (2011). A model, a heuristic and a decision support system to solve the scheduling problem of an earth observing satellite constellation. *Computers & Industrial Engineering*, 61(2):322–335.
- Wang, X., Wu, G., Xing, L., e Pedrycz, W. (2020). Agile earth observation satellite scheduling over 20 years: Formulations, methods, and future directions. *IEEE Systems Journal*, 15(3):3881–3892.
- Wolfe, W. J. e Sorensen, S. E. (2000). Three scheduling algorithms applied to the earth observing systems domain. *Management Science*, 46(1):148–166.
- Zufferey, N., Amstutz, P., e Giaccari, P. (2008). Graph colouring approaches for a satellite range scheduling problem. *Journal of Scheduling*, 11:263–277.