



2024

Métodos heurísticos para otimização do problema de dimensionamento e sequenciamento de lotes

Fábio Ferreira MARTINS^a; Willy Alves de Oliveira SOLER^b; Rubia Mara de Oliveira SANTOS^b

^a Aluno de Graduação em Engenharia Civil, fabio.f@ufms.br

^b Professor(a) Doutor(a), willy.oliveira@ufms.br; rubia.oliveira@ufms.br

^a Faculdade de Engenharias, Arquitetura e Urbanismo e Geografia. Universidade Federal de Mato Grosso do Sul. Av. Costa e Silva, s/nº | Bairro Universitário | 79070-900 | Campo Grande, MS, Brasil.,

^b Instituto de Matemática. Universidade Federal de Mato Grosso do Sul. Av. Costa e Silva, s/nº | Bairro Universitário | 79070-900 | Campo Grande, MS, Brasil.

RESUMO

O presente trabalho aborda a aplicação de métodos heurísticos para resolver um problema de planejamento de produção capacitado com dependência de sequência nos custos de *setup*, pedidos indivisíveis e janelas de tempo de entrega. A pesquisa investiga diferentes abordagens baseadas em busca local e propõe uma regra de simplificação do problema, que diminuem significativamente o tempo computacional de busca por uma solução de boa qualidade. Foram testados diferentes parâmetros e configurações para otimizar a performance dos métodos estudados. Os resultados mostram que a combinação de heurísticas construtivas com métodos de busca local iterativa ou mesmo a redução do problema original, proporcionam melhorias significativas na qualidade das soluções, minimizando o gap entre o limite dual e a solução viável.

Palavras-chave: Otimização, Heurística, Dimensionamento de Lotes, Sequenciamento, Pesquisa Operacional

ABSTRACT

In this work, we apply several heuristic approaches to deal with a production planning problem considering the Capacitated Lot Sizing Problem with Sequence-Dependent Setup costs, indivisible orders, and delivery time windows, more specifically, different local search-based approaches and a simplification rule, that significantly reduces the computational time required to find a high-quality solution. Various parameters and configurations were tested to optimize the performance of these methods. The results show that combining constructive heuristics with iterative local search methods, or even reducing the original problem, provides significant improvements in solution quality by minimizing the gap between the dual bound and the feasible solution.

Keywords: Optimization, Heuristics, Lot sizing, Scheduling, Operations Research

1. INTRODUÇÃO

Otimizar o processo de planejamento de produção é um desafio de otimização complexo, que surge em diversos contextos de manufatura e gestão da cadeia de suprimentos. Na construção civil, especificamente na indústria de elementos pré-fabricados de concreto, o problema de dimensionamento e sequenciamento de lotes surge quando há um *trade-off* significativo entre realizar várias trocas na linha de produção, a fim de atender um rol de demandas personalizadas ou realizar poucas trocas na linha de produção, deixando de atender

algumas demandas, mas minimizando custos associados às trocas na linha de produção e estocagem e maximizando o lucro proveniente das demandas que forem aceitas. O problema se torna mais complexo quando todos os pedidos devem entregues, integralmente, em janelas de tempo de entrega específica pré-determinada pelo cliente.

Neste estudo, consideramos um horizonte de planejamento finito, pedidos indivisíveis com a restrição de janela de tempo de entrega, bem como restrições de capacidade de produção, custos de manutenção de inventário e custos de *setup* sempre que houver troca na linha de produção. No entanto, há

flexibilidade para aceitar ou rejeitar pedidos, a fim de maximizar os lucros.

Para abordar este problema, propomos uma abordagem MIP-Heurística baseada em uma regra de simplificação que reduz o espaço de busca por soluções, mas mantendo sua variabilidade. Além disso, foram aplicados dois métodos heurísticos baseados em busca local. Especificamente, examinamos duas heurísticas de melhoria: algoritmos de busca local iterativa (*Diversify, Refine & Tight-Refine/DRT*) e de recozimento simulado (*Simulated Annealing/SA*).

2. REVISÃO DA LITERATURA

O Problema de Dimensionamento de Lotes (LSP) é um problema de otimização que busca determinar a quantidade ideal de produtos a serem produzidos em cada período com o objetivo de minimizar o custo total associado ao processo produtivo (Wagner & Whitin, 1958). Para atender à demanda de produtos, o objetivo é encontrar o tamanho ideal do lote que minimize os custos do processo produtivo (Medeiros *et al.*, 2022). É importante notar que o LSP é um problema comum na indústria, uma vez que, segundo Karmarkar e Schrage (1985), a produção em larga escala tende a ser mais econômica do que a produção em pequena escala, uma vez que os custos do processo produtivo são mais diluídos.

O modelo *Economic Order Quantity* (EOQ), proposto por Ford Whitman Harris em 1913, é uma abordagem clássica no planejamento da produção que busca determinar a quantidade ideal de produção ou de pedido para minimizar os custos totais associados à produção e ao estoque. Esse modelo considera um cenário simplificado, em que há apenas um único produto sendo produzido em uma linha de produção com capacidade ilimitada, assumindo também que o horizonte de planejamento é infinito e que as demandas são previamente conhecidas. Apesar de sua relevância histórica e teórica, o EOQ apresenta limitações para lidar com situações mais realistas. Por isso, ao longo dos anos, novos modelos e extensões foram desenvolvidos para incorporar características práticas e ampliar a aplicabilidade desse conceito.

O algoritmo de Wagner e Whitin, apresentado em 1958, é uma abordagem de programação dinâmica desenvolvida para resolver o problema do *Economic Order Quantity* (EOQ). Este método determina a solução ótima do problema com tempo de execução computacional polinomial, o que representa um

avanço significativo em termos de eficiência. A principal ideia do algoritmo está na caracterização das soluções ótimas, demonstrando que na ausência de restrições de capacidade, a produção deve ocorrer apenas quando o estoque estiver esgotado, de forma a minimizar os custos de preparação e armazenamento. Essa estratégia permitiu um avanço significativo na solução de problemas relacionados à gestão de estoques e planejamento de produção.

Um elemento central do algoritmo é sua habilidade de equilibrar o *trade off* entre os custos de estoque e os custos de *setup*. Manter produtos em estoque por períodos mais longos aumenta os custos de armazenagem, enquanto manter o estoque baixo pode significar uma dinamização maior da linha de produção, podendo aumentar os custos associados à produção.

O modelo *Capacitated Lot Sizing Problem* (CLSP), proposto por Trigeiro *et al.* (1989), introduziu uma formulação do problema de dimensionamento de lotes que incorporava restrições de capacidade de produção, tempos e custos de preparação variáveis entre períodos, além do conceito de *setup carryover*, que considerava a continuidade do estado de preparação da linha de produção entre períodos adjacentes.

Para abordar o problema, foi utilizada uma abordagem baseada em relaxação Lagrangeana, onde as restrições de capacidade foram dualizadas, permitindo a decomposição do modelo em subproblemas mais simples e não capacitados para cada produto individual. Esses subproblemas foram resolvidos utilizando o algoritmo de Wagner e Whitin (1958), e um procedimento heurístico foi desenvolvido para garantir a obtenção de soluções factíveis em um tempo computacional reduzido.

O modelo *Discrete Lot Sizing and Scheduling Problem* (DLSP), proposto por Fleischmann (1990), tratava do planejamento da produção de múltiplos produtos em uma única linha de produção, considerando uma demanda dinâmica e determinística que deve ser atendida sem atrasos. O objetivo do modelo é minimizar os custos associados ao estoque e à preparação da produção. O horizonte de planejamento é dividido em períodos discretos, como dias, semanas ou meses, dependendo do contexto. No DLSP, adota-se a política de "tudo ou nada", em que, durante um período, se a linha é preparada para produzir um produto específico, toda a sua capacidade produtiva é alocada para maximizar a produção desse item.

Posteriormente, Fleischmann (1994), expandiu o modelo DLSP para incluir custos de preparação dependentes da sequência, tornando o problema mais próximo de aplicações industriais reais. Essa reformulação foi inspirada no problema do caixeiro viajante com janelas de tempo (TSPTW), o que permitiu capturar as complexidades da sequência de produção. Além disso, foram propostos métodos baseados em relaxação Lagrangeana para calcular limites inferiores e heurísticas para a obtenção de soluções factíveis.

O modelo *Discrete Lot Sizing and Scheduling Problem* (DLSP) foi adaptado por Drexel e Haase, em 1995, para superar a limitação imposta pela política de "tudo ou nada". A partir dessa adaptação, surgiu o modelo *Proportional Lotsizing Problem* (PLSP), que permite a produção de até dois diferentes tipos de produtos em cada período produtivo, oferecendo maior flexibilidade na alocação da capacidade produtiva.

Além disso, o PLSP incorpora a preservação do estado de preparação da linha de produção (*setup carryover*), garantindo continuidade entre períodos consecutivos. Nesse contexto, o primeiro item a ser produzido em um período deve ser o mesmo que estava em produção no final do período anterior. Essa abordagem torna o modelo mais aderente a cenários industriais reais, onde é comum aproveitar configurações previamente estabelecidas para reduzir custos e tempos de *setup*.

O modelo *Capacitated Lotsizing Problem with Sequence Dependent Setup Costs* (CLSD), proposto por Haase em 1996, aborda o problema de planejamento da produção considerando limitações de capacidade produtiva e custos de preparação dependentes da sequência de produção. Esse modelo representa uma importante generalização do *Proportional Lotsizing Problem* (PLSP), permitindo a produção de diversos tipos de produtos em um único período produtivo. O CLSD incorpora o conceito de *setup carryover*, preservando o estado de preparação da linha entre períodos consecutivos, além de possibilitar a determinação da sequência ideal de produção dos itens. O objetivo principal é minimizar a soma dos custos de preparação e de armazenamento de estoques, o que torna o modelo mais aplicável a cenários industriais com demanda dinâmica e múltiplos produtos, onde a eficiência operacional e a redução de custos são essenciais.

James e Almada-Lobo (2011) exploraram o *Capacitated Lotsizing Problem with Sequence Dependent Setup Costs* (CLSD) em um ambiente com múltiplas linhas de produção paralelas. Para abordar

essa complexidade, os autores desenvolveram um procedimento heurístico denominado *Iterative Neighborhood Search* (INS), composto por duas fases principais: uma fase inicial de busca local e uma fase de modificação de vizinhança. A segunda fase é semelhante à técnica *Variable Neighborhood Search* (VNS). Os resultados obtidos demonstraram a eficácia do INS, fornecendo soluções de alta qualidade tanto para cenários envolvendo uma única linha de produção quanto para ambientes com várias linhas paralelas.

Soler, Poldi e Santos (2019) investigaram um problema de dimensionamento de lotes (PDL) aplicado à indústria de carnes. Nesse contexto, as demandas são representadas por pedidos indivisíveis que devem ser completamente aceitos e produzidos ou rejeitados integralmente. Além disso, os pedidos aceitos devem ser entregues em um único período dentro de uma janela de tempo estipulada pelos clientes. Os autores ampliaram o modelo *Capacitated Lotsizing Problem with Sequence Dependent Setup Costs* (CLSD) para incorporar essas novas restrições e características.

Para resolver o problema, foi proposto um procedimento heurístico composto por três fases: construção, melhoria determinística e melhoria estocástica. Os experimentos computacionais demonstraram que a heurística desenvolvida supera o desempenho do algoritmo *Branch-and-Bound* de um *solver* comercial de última geração, fornecendo soluções de qualidade superior para o problema abordado.

O estudo de Medeiros *et al.* (2022) apresenta um modelo de dimensionamento de lotes para atender demandas complexas na indústria alimentícia, onde o produto deve ser mantido em condições refrigeradas durante o armazenamento e a produção devido à perecibilidade. O modelo proposto é capaz de minimizar o custo total de produção, levando em conta fatores como tempo de processamento, custo de armazenamento, bem como restrições de capacidade de produção e disponibilidade de recursos escassos.

Copil *et al.* (2016) conduziram uma importante revisão histórica do problema de dimensionamento e programação de lotes simultâneos (LSP). Os autores também propuseram um esquema de classificação para os modelos e identificaram tendências tanto na academia quanto na indústria em relação ao LSP.

Ceschia *et al.* (2017) propuseram uma abordagem híbrida para resolver o problema de dimensionamento e programação de lotes,

combinando a heurística *Simulated Annealing* (SA) com o método exato *Branch and Bound* de um *solver* comercial. A heurística SA é baseada em conceitos probabilísticos e visa aproximar o ótimo global de uma função (Cerny, 1985, e Kirkpatrick *et al.*, 1983).

2.1 O modelo CLSD-DTI

O problema compreende a otimização das tomadas de decisões relacionadas à produção e entrega em um ambiente industrial, com o objetivo de maximizar o lucro total. Entre suas características principais estão os custos de manutenção de estoque, e os custos e tempos de setup dependentes da sequência de produção, que variam conforme a ordem em que os itens são fabricados. Além disso, o modelo considera a flexibilidade na aceitação dos pedidos, onde é possível determinar quais pedidos serão aceitos, levando em conta restrições de capacidade e o objetivo de maximizar o lucro. Outro aspecto relevante é a indivisibilidade dos pedidos. Nesse contexto, caso o pedido seja aceito, ele deve ser entregue, por inteiro, em um único período. O problema também incorpora janelas de tempo de

entrega, que estabelecem intervalos específicos para a entrega dos pedidos.

A função objetivo (1) visa minimizar os custos de estoque e de setup, e maximizar o lucro obtido por cada pedido que for aceito. O balanço de inventário é regulado pela Restrição (2).

A Restrição (3) garante que cada pedido seja atendido no máximo uma vez, dentro de sua janela de tempo designada. As limitações de capacidade são impostas pela Restrição (4), e a Restrição (5) estipula que a produção de um produto está condicionada ao setup da linha de produção para este item.

A Restrição (6) garante o sequenciamento de todos os itens produzidos dentro do período correspondente. A Restrição (7) dita que há um único primeiro item produzido em cada período.

A Restrição (8) mantém o balanço do fluxo de sequenciamento para os lotes, e a Restrição (9), conhecida como restrição MTZ, é implementada para prevenir *sub-tours* no problema de sequenciamento.

As restrições (10) e (11) dispõem sobre o domínio das variáveis de decisão.

As Tabelas 1 e 2 apresentam os parâmetros do modelo e as variáveis de decisão, respectivamente.

Tabela 1 - Parâmetros. Fonte: Autor

Parâmetros	
N, T	Número de pedidos (n) e número de períodos (t)
J	Número de itens (i e j)
F_n, L_n	Primeiro e último período da janela de tempo de entrega do pedido n
P_{nt}	Lucro do pedido n, se entregue no período t
O_t	Conjunto de pedidos que podem ser entregues no período t
q_{jn}, C_t	Quantidade do item j no pedido n e capacidade de produção do período t
h_j, a_j	Custo unitário de armazenagem e tempo unitário de processamento do item j
sc_{ij}, st_{ij}	Custo de setup e tempo de setup para <i>setup</i> do produto i para o produto j

Tabela 2 - Parâmetros. Fonte: Autor

Variáveis de decisão	
γ_{nt}	1 se o pedido n é entregue no período t , e 0 caso contrário
w_{jt}	1 se o item j é produzido no período t , e 0 caso contrário
y_{jt}	1 se o item j é o primeiro item produzido no período t , e 0 caso contrário
z_{ijt}	1 se há mudança de produção do item i para o item j no período t , e 0 caso contrário
x_{jt}	Quantidade do item j produzida no período t
I_{jt}	Inventário do item j ao final do período t
V_{jt}	Variáveis auxiliares para representar a ordem de produção do item j no período t

A formulação matemática do modelo consiste em:

$$\max \sum_{n=1}^N \sum_{t=F_n}^{L_n} P_{nt} \gamma_{nt} - \sum_{t=1}^T \sum_{j=1}^J h_j I_{jt} - \sum_{t=1}^T \sum_{i=1}^J \sum_{j=1}^J sc_{ij} z_{ijt} \quad (1)$$

$$I_{j,t-1} + x_{jt} = I_{jt} + \sum_{n \in O_t} q_{jn} \gamma_{nt} + I_{jt} \quad \forall j, t; \quad (2)$$

$$\sum_{t=F_n}^{L_n} \gamma_{nt} \leq 1 \quad \forall n; \quad (3)$$

$$\sum_{j=1}^J a_j x_{jt} + \sum_{i=1}^J \sum_{j=1}^J st_{ij} z_{ijt} \leq C_t \quad \forall t; \quad (4)$$

$$x_{jt} \leq \frac{C_t}{a_j} w_{jt} \quad \forall j, t; \quad (5)$$

$$w_{jt} = y_{jt} + \sum_{i=1}^J z_{ijt} \quad \forall j, t; \quad (6)$$

$$\sum_{j=1}^J y_{jt} = 1 \quad \forall t; \quad (7)$$

$$y_{jt} + \sum_{i=1}^J z_{ijt} = \sum_{i=1}^J z_{jit} + y_{j,t+1} \quad \forall j, t; \quad (8)$$

$$V_{jt} \geq V_{it} + 1 - J(1 - z_{ijt}) \quad \forall i, j, t; \quad (9)$$

$$\gamma_{nt} \in \{0,1\}; \quad \forall n, t; \quad (10)$$

$$w_{jt}, y_{jt} \in \{0,1\}; \quad \forall j, t; \quad (11)$$

$$z_{ijt} \in \{0,1\}; \quad \forall i, j, t; \quad (12)$$

$$x_{jt}, I_{jt}, V_{jt} \geq 0; \quad \forall j, t; \quad (13)$$

2.1.1 Métodos de solução

No modelo CLSD-DTI, trabalha-se com a hierarquia de variáveis, o que permite a priorização de certas variáveis sobre outras, potencialmente levando a soluções mais eficientes em termos de tempo computacional. Uma variável crítica nesse modelo é " γ_{nt} ", que representa se o pedido 'n' será entregue no período 't'. Fixar o valor desta variável limita o espaço de busca das demais variáveis, reduzindo significativamente as possibilidades combinatórias do problema com um todo. Métodos heurísticos exploram essa hierarquia de variáveis para obter uma solução factível num tempo menor. Notavelmente, métodos abordados na presente pesquisa, como o DRT (Fischetti et al., 2004) e o *Simulated Annealing* (SA) (Cerny, 1985), ambos utilizando a regra de ramificação local de Fischetti e Lodi (2003), aproveitam este conceito de hierarquia de variáveis. Também foi utilizada uma heurística do tipo *relax-and-fix* que se mostrou capaz de fornecer soluções factíveis em tempo computacional aceitável. Estas soluções foram utilizadas como ponto de partida para os demais métodos estudados.

Além disso, uma nova MIP-Heurística proposta neste estudo visa restringir o espaço de busca do modelo direcionando (mas não fixando) a maioria das variáveis " γ_{nt} ", mas ainda garantindo que uma boa solução possa ser encontrada pelo *solver* MIP.

2.1.2 Heurística construtiva Relax-and-fix (RF)

O procedimento *Relax-and-Fix* (RF) é uma heurística construtiva iterativa desenvolvida para encontrar soluções viáveis para problemas de programação inteira mista, particularmente aplicável a problemas de planejamento da produção com horizonte temporal subdividido em múltiplos

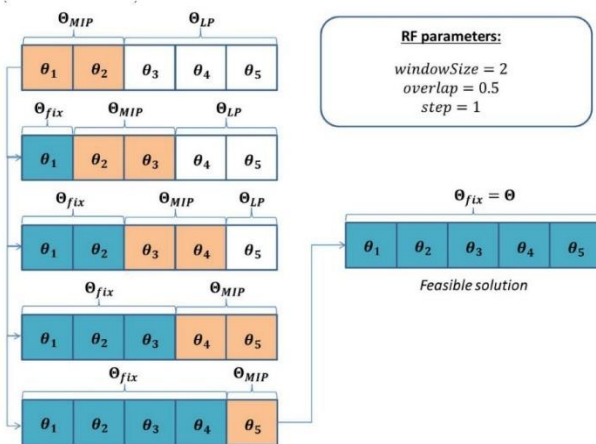
períodos. O objetivo é decompor o conjunto de variáveis binárias em subconjuntos menores associados a períodos específicos, facilitando a solução do problema em etapas. Inicialmente, todas as variáveis binárias são relaxadas, ou seja, tratadas como contínuas, para simplificar o problema. Em seguida, define-se uma janela (*window*) de tamanho fixo, determinado pelo parâmetro *windowSize*, que abrange um número limitado de períodos contíguos. As variáveis binárias correspondentes aos períodos dentro dessa janela são tratadas como inteiras e resolvidas pelo *solver*, enquanto as variáveis fora da janela permanecem relaxadas.

Após a resolução do subproblema correspondente à janela inicial, os valores das variáveis que deixaram a janela (devido ao deslocamento da mesma) são fixados em suas soluções (ótimas ou não), formando o conjunto Θ_{fix} . Em seguida, a janela é deslocada para frente de acordo com o passo definido pelo parâmetro *step*, que é calculado como $((1-overlap) \times windowSize)$, onde *overlap* representa a sobreposição entre janelas consecutivas. Esse processo iterativo de mover a janela, resolver o subproblema e fixar variáveis é repetido até que todas as variáveis binárias tenham sido tratadas como inteiras pelo menos uma vez, momento em que a solução final é obtida.

O RF possui algumas vantagens importantes. Primeiro, ele simplifica o problema como um todo, ao trabalhar com subproblemas menores a cada iteração, o que torna viável a resolução de modelos de grande escala. Em segundo lugar, a flexibilidade do RF permite explorar diferentes configurações de parâmetros, como o tamanho da janela, o nível de sobreposição (*overlap*) e o limite de tempo de execução (*timeLimit*), para adaptar o algoritmo a diferentes cenários. Por fim, a suposição de flexibilidade na aceitação de demandas (como no caso do CLSP-DTI) garante que, mesmo em situações de alta restrição, o algoritmo possa encontrar uma solução ao permitir a rejeição de pedidos inviáveis.

Figura 2 - Esquema do procedimento "RF" considerando $T = 5$, $windowSize = 2$, $overlap = 0.5$, e $step = (1 - overlap) * windowSize = 1$.

Fonte: Soler, Poldi e Santos (2019)



Abaixo, segue o Pseudocódigo 1, da execução do algoritmo *relax-and-fix*.

Pseudocódigo 1 – Algoritmo da heurística *Relax-and-fix*.
Fonte: Autor

Algoritmo Relax-and-Fix (RF)

```

set parameters windowSize, overlap, timeLimit
inicializar  $\Theta = (\theta_1, \theta_2, \dots, \theta_T)$ 
initialize sol. $\Theta$  as LinearlyRelax( $\Theta$ )

enquanto  $\Theta_{fix} \neq \Theta$  do
    se a primeira interação
         $\Theta_{MIP} \leftarrow \text{DefineWindow}(\Theta, \text{windowSize})$ 
         $\Theta_{LP} \leftarrow \Theta \setminus \Theta_{MIP}$ 
    else
         $\Theta_{fix} \leftarrow \text{UpdateFixedVariables}(\Theta, \text{previousWindow})$ 
         $\Theta_{MIP} \leftarrow \text{DefineWindow}(\Theta, \text{windowSize}, \Theta_{fix})$ 
         $\Theta_{LP} \leftarrow \Theta \setminus (\Theta_{MIP} \cup \Theta_{fix})$ 

sol. $\Theta \leftarrow \text{SolveMIP}(\Theta_{MIP}, \Theta_{LP})$ 
UpdateWindowParameters(windowSize, overlap)
    
```

2.1.3 MIP-Heurística

O método em questão representa uma abordagem simples e direta para resolver o problema

CLSD-DTI. Seu objetivo é restringir o espaço de busca boa parte das variáveis “ γ_{nt} ”. A estratégia baseia-se na introdução de uma nova restrição no modelo original, que determina que todos os pedidos que forem aceitos devem ser entregues exclusivamente no primeiro ou no último período de sua janela de tempo designada, excluindo assim, a possibilidade de um pedido ser entregue num período intermediário de sua janela de tempo de entrega.

Essa restrição reduz a vastamente as possibilidades combinatórias do problema, pois limita o número de configurações possíveis para as variáveis “ γ_{nt} ” na solução.

Apesar dessa simplificação, a abordagem não compromete a factibilidade da solução. Ou seja, a solução ótima encontrada para o modelo simplificado pode não coincidir com a solução ótima do problema original, mas ela é sempre viável, uma vez que respeita as restrições de tempo de entrega e as demais condições impostas pelo problema.

Essa abordagem é promissora em cenários onde a busca por soluções de qualidade em tempo viável é crítica, e onde a redução das possibilidades combinatórias é essencial para lidar com instâncias grandes do problema.

2.1.4 Heurísticas de melhora

2.1.4.1 DRT

O método *Diversify, Refine & Tight-refine* (DRT), proposto por Fischetti *et al.* (2004) é um processo iterativo de busca em vizinhança dividido em 3 fases: (i) Diversificação; (ii) Refinamento; e (iii) Refinamento-Fino.

Cada uma dessas fases utiliza parâmetros específicos para definir a vizinhança de soluções permitidas, buscando equilibrar a exploração de diferentes regiões do espaço de busca e a intensificação em regiões promissoras.

Inicialmente, as variáveis binárias são subdivididas em variáveis de primeiros ou segundo nível. No nosso caso, as variáveis de primeiro nível são as variáveis “ γ_{nt} ” associadas a pedidos cuja janela de tempo de entrega seja de um único período, enquanto as variáveis de segundo nível englobam todas as demais variáveis “ γ_{nt} ”.

Neste contexto, a vizinhança de uma solução é definida como o conjunto de soluções que podem ser alcançadas ao diferir da solução atual

(incumbente) em até um número limitado de variáveis “ γ_{nt} ”. Essa definição é central para o método e é controlada pelos seguintes parâmetros:

- **$\Delta 1$** : delimita a vizinhança de soluções permitidas para alterações nas variáveis de primeiro nível.
- **$\Delta 2$** : delimita a vizinhança de soluções permitidas para alterações nas variáveis de segundo nível.
- **$k1$** : quantidade de variáveis de primeiro nível que podem ser alteradas. Valores menores restringem a vizinhança, enquanto valores maiores permitem maior flexibilidade.
- **$k2$** : quantidade de variáveis de segundo nível que podem ser alteradas. Este parâmetro surge apenas na fase de Refinamento-fino.

Em seguida, durante a fase de refinamento, a busca local é restrita às variáveis de primeiro nível. Esta fase incorpora a restrição (14) ao modelo. Aqui, são permitidas alterações para as variáveis de primeiro nível, desde que não ultrapasse o limite $k1$, em geral um número entre 6 e 10. Foram realizados testes para vários valores de $k1$.

$$\Delta 1 \leq k1 \quad (14)$$

Se o mínimo local não for encontrado na fase de refinamento, passamos para a fase de refinamento-fino. Este processo é caracterizado pela introdução da restrição (15), agora para as variáveis de segundo

nível. Desta forma, a vizinhança $\Delta 2$ é definida como o espaço de soluções, cujas variáveis de segundo nível diferem da solução incumbente em $k2$ (em geral, $k2$ inicia-se em um número pequeno)

$$\Delta 2 = k2 \quad (15)$$

Cada vez que o mínimo local é atingido, $k2$ é incrementado gradualmente em 2. É recomendável que os parâmetros $k1$ e $k2$ sejam números pares, pois a transição de uma solução para outra geralmente envolve aceitar um pedido enquanto rejeita-se outro. Se o mínimo local não for encontrado dentro do tempo-limite desta fase, é dado prosseguimento para a próxima fase: a diversificação.

A diversificação visa expandir a busca para outras vizinhanças de soluções, permitindo maior variação. Nesta fase, as restrições das fases anteriores são removidas e uma nova restrição é introduzida no modelo:

$$k1_{min} \leq \Delta 1 \leq k1_{max} \quad (16)$$

Onde $\Delta 1$ denota a vizinhança de soluções associadas a mudanças das variáveis de primeiro nível, restritas ao *range* de $k1_{min}$ a $k1_{max}$. O modelo é então submetido a um *solver*, e qualquer solução identificada é considerada como critério de parada.

O pseudocódigo 2 esquematiza a execução do DRT:

Pseudocódigo 2 – Algoritmo da heurística DRT. Fonte: Autor

Algoritmo DRT

Solução ← Primeira solução encontrada (MIP_Model)

Enquanto (Tempo Atual < Tempo Limite):

 Nova Solução ← Solver (MIP com restrição de diversificação (16))

 Remova a restrição de diversificação

 Solução Local ← Solver (MIP com restrição de Refino (14))

Se (Solução local não é ótima):

Enquanto ($k2 < k2_{max}$):

 Nova Solução ← Solver (MIP com restrição de Refinamento fino (15))

Se ($f(\text{Nova Solução}) < f(\text{Solução})$):

 Solução ← Nova Solução

$k2 = k2 + k2_step$

 Remova todas as restrições temporárias

Retornar Solução

2.1.4.2 Recozimento simulado (*Simulated Annealing*)

O *Simulated Annealing* (SA) é uma técnica de otimização inspirada no processo metalúrgico de têmpera, onde materiais são aquecidos e depois resfriados para reduzir defeitos, alcançando assim um estado de energia mínima. A ideia central do SA é simular esse processo de resfriamento computacionalmente, permitindo que a busca escape de mínimos locais e se mova em direção ao mínimo global da função objetivo. Este método precisa das seguintes etapas:

Inicialização: começar com uma solução inicial e uma “temperatura inicial” alta.

Iteração: a cada passo do algoritmo, uma nova solução é gerada fazendo uma pequena alteração na solução atual. No nosso caso, o movimento de uma solução para outra é definido pela regra de ramificação local proposta por Fischetti & Lodi (2003), que também está presente na fase de Refinamento (Restrição 14).

Critério de Aceitação: a nova solução é aceita de acordo com o critério de Metropolis: se a nova solução melhora a função objetivo, ela é sempre aceita. Se a nova solução não melhora a função objetivo, ela tem a possibilidade de ser aceita, se satisfizer o critério abaixo:

$$e^{\frac{-\Delta}{T}} > \text{randomiza}(0,1) \quad (17)$$

Cronograma de Resfriamento: a temperatura é gradualmente diminuída de acordo com um cronograma de resfriamento. No nosso caso, o cronograma de resfriamento é determinado pela regra simples de que, a cada iteração, a temperatura é igual a 80% da temperatura anterior.

Término: o algoritmo termina após o alcance de um limite de tempo, quando a temperatura atinge um determinado limite inferior ou quando não há melhoria significativa na qualidade da solução observada ao longo de um determinado número de etapas. A melhor solução encontrada durante o processo é retornada como saída.

O Pseudocódigo 3 descreve o procedimento da heurística SA.

Pseudocódigo 3 – Algoritmo da heurística *Simulated Annealing* (SA). Fonte: Autor

Pseudo-Código Procedimento SA

Solução ← Primeira solução encontrada(MIP_Model)

Enquanto (temperatura > Temp_mínima e Tempo Limite não atingido):

 Nova solução ← Busca local (Eq. 14)

 Diferença ← [f (Nova solução) – f (Solução)]

Se [Diferença < 0] **ou** [Random(0,1) < exp(-Diferença / temperatura)] :

 Solução ← Nova solução

 temperatura ← temperatura * 0,8

Retornar Solução

Ao explorar iterativamente o espaço de soluções e aceitar soluções com base em um critério probabilístico, este procedimento combate o problema de ficar preso em ótimos locais.

3. RESULTADOS COMPUTACIONAIS

3.1 Instâncias de teste

Para avaliar a eficácia das abordagens propostas, foi utilizado um conjunto de 100 instâncias de teste presentes na literatura. (Soler, Poldi e Santos. 2019), com uma ampla gama de cenários. As instâncias de teste foram divididas em dez classes, cada uma contendo dez instâncias, com base nas escolhas de parâmetros.

A notação N30 J15 T5, por exemplo, é usada para indicar a classe de instâncias com N=30 pedidos, J=15 itens disponíveis e T=5 períodos.

3.2 Ambiente de teste

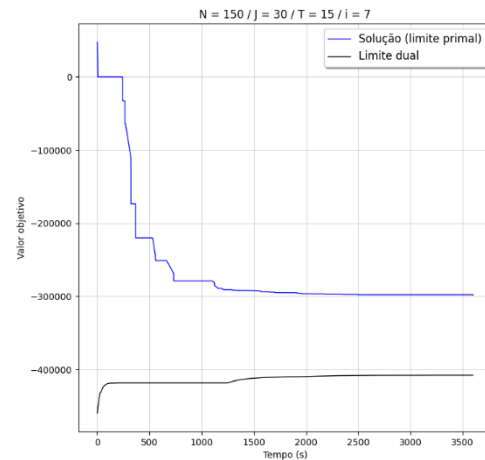
O modelo proposto e os procedimentos heurísticos foram implementados na linguagem de programação *Python*, usando a biblioteca *Python-MIP* (Santos & Toffolo, 2019) para conectar a linguagem de programação ao *solver* comercial estado da arte Gurobi 9.1.1. Os subproblemas MIP foram resolvidos pelo algoritmo *Branch-and-Bound* do *solver* usando as configurações padrão. Os testes foram conduzidos em um *desktop* com dois processadores *Intel Xeon* a 2,8 GHz e 128 GB de RAM DDR3. Para cada instância, foram registrados a melhor solução viável (z^f) e o melhor limite dual (z^d). O desvio da melhor solução viável em relação ao limite inferior (GAP) foi calculado, conforme a equação (18):

$$GAP = 100 * \frac{z^d - z^f}{z^f} \quad (18)$$

Frisa-se que nesta seção, todos os valores de gap foram calculados utilizando o melhor limite dual obtido pelo *solver* Gurobi para o problema original.

A Figura 2 exemplifica a execução do algoritmo *Branch-and-Bound*. O problema dual é constantemente atualizado para fornecer um limite inferior, enquanto a solução viável melhora o limite superior, reduzindo progressivamente o *gap* entre os dois valores até convergir para a solução ótima.

Figura 2 – Exemplo de execução do algoritmo *Branch-and-Bound*. Fonte: Autor

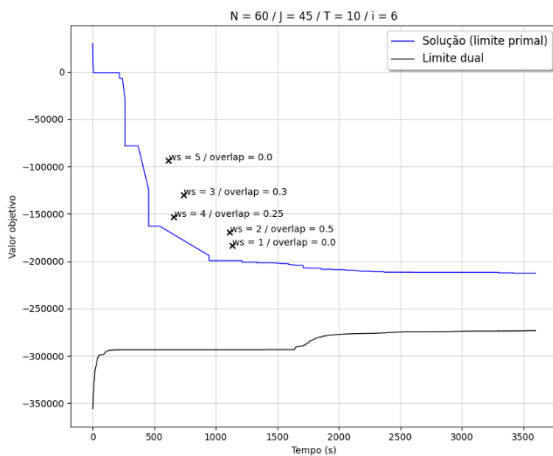


3.3 Comparação entre os métodos

Os métodos avaliados apresentam abordagens distintas para resolver o problema, com vantagens que refletem na qualidade das soluções e nos tempos computacionais. Eles foram comparados em termos de qualidade de solução obtida e tempo computacional frente ao algoritmo de *Branch-and-Bound* do *solver* de última geração Gurobi versão 9.1.1. O *solver* serve como referência, fornecendo soluções ótimas ou limites inferiores, mas com um custo computacional elevado, especialmente para instâncias de maiores.

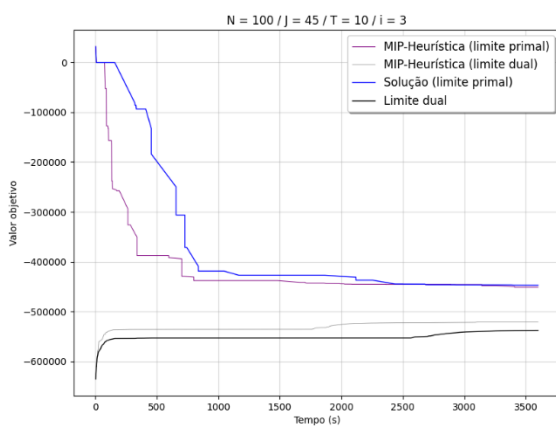
O *Relax-and-Fix* é um método baseado na divisão do problema em etapas menores. Apesar de ser menos exploratório, o RF é eficiente para grandes instâncias, destacando-se pela garantia de encontrar ao menos uma solução factível. Este método é usado para produzir “primeiras soluções” que são utilizadas como ponto de partida para os demais métodos investigados. A Figura 3, abaixo, exemplifica o impacto da escolha dos parâmetros do RF na qualidade da solução que ele produz para uma instância.

Figura 3 – Análise da execução do RF em diversos parâmetros para uma instância. Fonte: Autor



O método MIP-Heurístico proposto evita interações, introduzindo restrições que direcionam, de modo geral, as variáveis mais decisivas do problema. Nota-se que a qualidade das soluções obtidas foi, em geral, superior às obtidas pelo algoritmo *Branch-and-Bound*, do *solver* estado da arte. A Figura 4 mostra a execução da MIP-Heurística frente ao *solver*.

Figura 4 – Exemplo da execução da MIP-Heurística comparada ao *solver*. Fonte: Autor



O DRT combina busca em vizinhança e melhora iterativa, explorando o espaço de soluções em três fases. Na fase do refinamento, a busca é concentrada na vizinhança “ $\Delta 1$ ”, delimitada pelo parâmetro “ $k1$ ”; e, no refinamento fino, a vizinhança

é ainda mais restrita, delimitada pelo parâmetro “ $k2$ ”. Na fase de diversificação, amplia-se a vizinhança para explorar novos pontos, e evitar mínimos locais.

O SA simula um processo de resfriamento gradual para escapar de mínimos locais e buscar soluções próximas ao ótimo global. Por meio de um critério probabilístico, ele permite alterações temporariamente desfavoráveis para promover maior diversidade na exploração.

Embora seja altamente flexível e eficaz em grandes espaços de soluções, o método é sensível à parametrização (temperatura inicial, cronograma de resfriamento e tempo de execução). As Figuras 5 e 6 exemplificam a execução do DRT e SA, respectivamente.

Figura 5 – Comparativo. Fonte: Autor

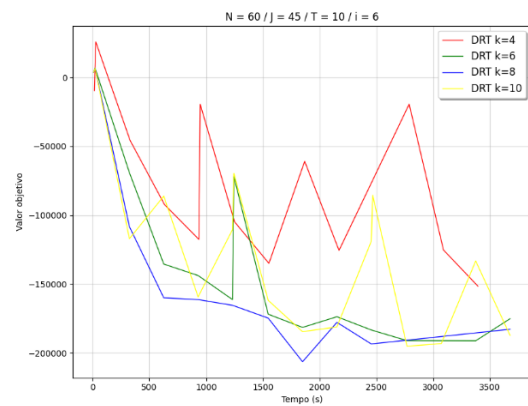
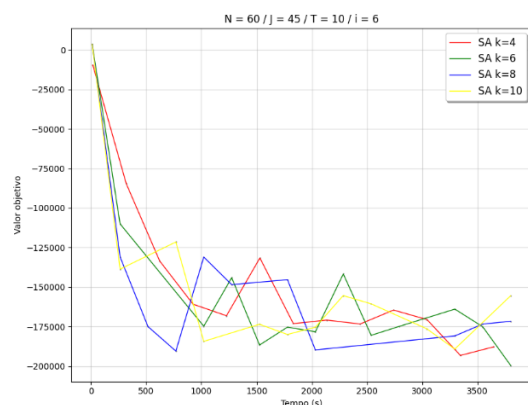


Figura 6 – Comparativo. Fonte: Autor



A Tabela 3 resume a comparação geral entre os métodos investigados, individualmente.

Tabela 3 – Resumo da comparação entre os métodos. Fonte: Autor

Método	Qualidade da Solução	Tempo Computacional	Aplicabilidade
<i>Relax-and-Fix</i> (RF)	Baixa	Baixo	Útil para grandes instâncias onde é necessário encontrar rapidamente uma solução factível, especialmente como ponto de partida para outros métodos.
MIP-Heurística	Alta (subótima)	Moderado	Adequado para problemas onde a qualidade da solução subótima é aceitável, mas com maior rapidez e simplicidade do que métodos iterativos.
<i>Diversify, Refine & Tight-refine</i> (DRT)	Moderado	Moderado	Ideal para problemas complexos onde a exploração detalhada do espaço de soluções é importante para evitar mínimos locais e obter soluções de alta qualidade.
<i>Simulated Annealing</i> (SA)	Moderado	Baixo	Indicado para problemas com grandes espaços de solução, onde é importante explorar diferentes regiões e escapar de mínimos locais, sendo necessário o ajuste mais sensível dos parâmetros.

A Tabela 4 apresenta o *gap* médio (%) obtido para os principais métodos estudados. Como referência, os resultados foram comparados com o *solver* Gurobi 9.1.1, utilizando o algoritmo *Branch-and-Bound*. O *Gap* foi calculado em relação ao

melhor limite inferior obtido para cada instância, avaliando o desempenho relativo dos métodos em diferentes classes de problemas. Para os métodos DRT e SA, os resultados são apresentados para diferentes valores do parâmetro “k”.

Tabela 4 – *Gap* médio entre os métodos. Fonte: Autor

Classe	<i>Solver</i> Gurobi 9.1.1	<i>Diversify, Refine & Tight-refine</i> (DRT)				<i>Simulated Annealing</i> (SA)				MIP-Heurística
		k1=4	k1=6	k1=8	k1=10	k=4	k=6	k=8	k=10	
N30 J15 T5	0,89	1,85	1,87	1,01	1,04	4,86	1,73	1,68	1,42	2,15
N30 J30 T5	12,88	12,42	11,75	11,95	13,09	14,19	13,14	13,35	13,53	10,55
N30 J45 T5	20,09	19,82	19,99	20,26	22,05	19,88	21,25	23,46	25,96	17,90
N50 J15 T5	3,47	3,76	3,41	3,68	3,30	8,49	3,77	3,73	3,51	3,77
N50 J30 T5	12,29	11,90	10,86	11,43	11,91	13,58	11,57	12,14	11,90	11,33
N50 J45 T5	15,74	15,94	15,08	15,59	15,60	15,73	16,04	16,37	17,12	15,02
N60 J45 T10	30,57	123,72	41,12	37,72	40,94	58,00	44,82	44,66	42,49	30,32
N100 J30 T10	20,78	42,66	30,92	25,38	26,87	34,92	25,27	25,39	22,93	19,51
N100 J45 T10	22,76	122,99	43,68	34,91	33,15	58,22	36,61	39,07	34,57	22,09
N150 J30 T15	32,67	117,47	77,03	57,70	53,38	73,57	63,41	51,95	42,14	32,58
Média	17,21	47,25	25,57	21,96	22,13	30,14	23,76	23,18	21,56	16,52

Já a Tabela 5, apresenta o desempenho do método *Relax-and-Fix* (RF) considerando diferentes combinações dos parâmetros *window size* (ws) e *overlap* (o). O ajuste desses parâmetros é essencial para o funcionamento do método, determinando a

divisão do problema em subproblemas menores e a sobreposição entre as janelas solucionadas. O percentual apresentado corresponde ao *gap* médio (%) em relação ao melhor limite inferior conhecido, para cada respectiva instância.

Tabela 5 – Desempenho do método *Relax-and-Fix*. Fonte: Autor

Classe	Window size = 1 overlap = 0	Window size = 2 overlap = 0.5	Window size = 3 overlap = 0.3	Window size = 4 overlap = 0.25	Window size = 5 overlap = 0
N30 J15 T5	14.83%	6.60%	4.34%	4.39%	3.97%
N30 J30 T5	19.36%	17.97%	18.89%	20.34%	17.16%
N30 J45 T5	25.93%	26.55%	23.71%	27.38%	24.63%
N50 J15 T5	9.55%	7.22%	6.96%	7.05%	6.41%
N50 J30 T5	14.35%	15.33%	14.98%	16.00%	14.79%
N50 J45 T5	17.46%	23.28%	18.22%	19.54%	18.13%
N60 J45 T10	46.64%	61.16%	124.79%	207.66%	180.77%
N100 J30 T10	21.99%	26.12%	42.68%	36.98%	35.62%
N100 J45 T10	32.69%	40.36%	70.44%	173.68%	134.45%
N150 J30 T15	66.17%	79.34%	83.97%	92.73%	131.33%
Média	26.90%	30.39%	40.90%	60.58%	56.72%

A Tabela 6 apresenta o *Gap* médio (%) obtido para todas as classes de instâncias, utilizando, como solução inicial, a solução obtida pelo *Relax-and-Fix* (RF). Para tanto, foram utilizados os parâmetros que produziram as soluções de melhor qualidade.

Essa abordagem demonstra a eficácia da estratégia de fornecer boas soluções iniciais, visto que houve melhora em todos os casos. Os valores são apresentados para as classes de instâncias, bem como a média geral para cada método na Tabela 7.

Tabela 6 – *Gap* médio obtido entre todas as classes as instâncias. Fonte: Autor

Método	Parâmetros
RF	$windowSize = 1, overlap = 0$
DRT	$k=8$
SA	$k=10$

Tabela 7 – Análise comparativa dos métodos estudados, a partir da solução inicial fornecida pelo RF. Fonte: Autor.

Classe	Solver	DRT (k=8)	SA (k=10)
N30 J15 T5	0.91%	1.43%	1.28%
N30 J30 T5	12.23%	12.49%	14.07%
N30 J45 T5	19.51%	21.65%	22.50%
N50 J15 T5	3.34%	3.07%	3.29%
N50 J30 T5	11.17%	10.91%	11.57%
N50 J45 T5	14.66%	14.89%	15.24%
N60 J45 T10	29.45%	39.43%	41.53%
N100 J30 T10	18.62%	20.84%	21.20%
N100 J45 T10	24.10%	30.43%	31.56%
N150 J30 T15	30.85%	42.54%	43.43%
Média	16.49%	19.77%	20.57%

Cumprir notar que a utilização do método *Relax-and-Fix* (RF) como solução inicial para a MIP-

Heurística requer ajustes específicos no procedimento do RF para atender às particularidades deste método.

Contudo, não é este o escopo desta pesquisa, dado que o objetivo principal foi avaliar o desempenho individual dos métodos e a qualidade das soluções obtidas a partir de suas implementações padrão.

4. CONSIDERAÇÕES FINAIS

A análise dos resultados permitiu uma avaliação abrangente do desempenho dos métodos investigados no contexto do problema. A partir dos dados apresentados, constatou-se que o *solver* Gurobi 9.1.1 obteve um bom desempenho, com um *Gap* médio de 17,21%, estabelecendo um parâmetro de referência para a comparação dos demais métodos. Esse desempenho é consistente com sua posição como um dos *solvers* mais rápidos do mundo.

O método MIP-Heurístico destacou-se como a abordagem com a melhor qualidade de solução, alcançando um *gap* médio de 16,52%, ligeiramente inferior ao do *solver*. Por sua vez, os métodos DRT e SA apresentaram desempenhos intermediários. O DRT alcançou um *gap* médio de 21,96% para $k=8$, enquanto o SA obteve um *gap* médio de 21,56% para $k=10$, sendo particularmente eficiente em instâncias de maior complexidade, embora com grande sensibilidade à parametrização.

A integração do método *Relax-and-Fix* (RF) como solução inicial nos métodos DRT e SA demonstrou vantagens consideráveis. Conforme evidenciado pelos dados, as soluções iniciais geradas pelo RF reduziram o *gap* médio em ambos os casos. Em particular, o *gap* médio do DRT foi reduzido de 21,96% para 19,77%, enquanto o do SA foi aprimorado de 21,56% para 20,57%.

Para pesquisas futuras, recomenda-se o aprimoramento da integração entre os métodos, com foco em abordagens híbridas, visto que a utilização de uma solução inicial como ponto de partida se mostrou eficaz na redução do *gap* médio. Além disso, a exploração de novas técnicas de modelagem que incorporem de forma eficiente a crescente disponibilidade de dados, impulsionada pela Indústria 4.0, surge como um caminho promissor. Essas abordagens têm potencial para aumentar a qualidade das soluções e expandir a aplicabilidade dos métodos heurísticos para cenários mais diversificados.

5. AGRADECIMENTOS

À minha família, a quem eu dedico este trabalho, pelo apoio incondicional. Ao meu orientador e aos demais os professores que contribuíram para minha formação. À minha namorada, Priscila, pelo incentivo constante. Ao curso de Engenharia Civil e à Universidade Federal de Mato Grosso do Sul pelo suporte e oportunidades proporcionadas ao longo desta jornada acadêmica. Ao Centro de Ciências Matemáticas Aplicadas à Indústria (CeMEAI), pelos recursos computacionais. Ao Centro Acadêmico de Engenharia Civil, pela oportunidade de contribuir ativamente na organização dos estudantes. Por fim, a todos os colegas e amigos que, direta ou indiretamente, colaboraram para o alcance deste objetivo.

REFERÊNCIAS BIBLIOGRÁFICAS

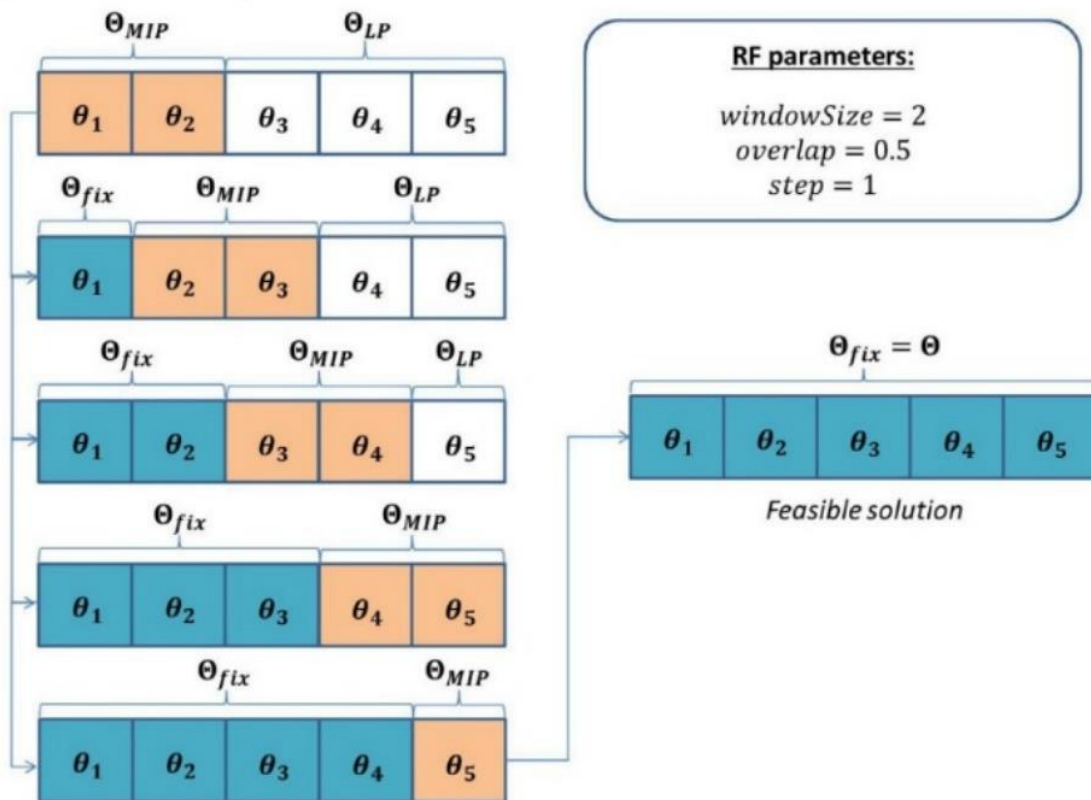
- CESCHIA, S.; DI GASPERO, L.; SCHAERF, A. **Solving discrete lot-sizing and scheduling by simulated annealing and mixed integer programming**. *Computers & Industrial Engineering*, v. 114, p. 235-243, 2017. Disponível em: <https://doi.org/10.1016/j.cie.2017.10.017>.
- COFIL, K.; WÖRBELAUER, M.; MEYR, H.; TEMPELMEIER, H. **Simultaneous lotsizing and scheduling problems: a classification and review of models**. *OR Spectrum: Quantitative Approaches in Management*, v. 39, n. 1, p. 1-64, 2017. Disponível em: https://EconPapers.repec.org/RePEc:spr:orspec:v:39:y:2017:i:1:d:10.1007_s00291-015-0429-4.
- DREXL, A.; HAASE, K. **Proportional lotsizing and scheduling**. *International Journal of Production Economics*, v. 40, n. 1, p. 73-87, 1995. Disponível em: [https://doi.org/10.1016/0925-5273\(95\)00040-U](https://doi.org/10.1016/0925-5273(95)00040-U).
- FISCHETTI, M.; LODI, A. **Local branching**. *Mathematical Programming*, v. 98, n. 1, p. 23-47, 2003. Disponível em: <https://doi.org/10.1007/s10107-003-0395-5>. Acesso em: 3 jan. 2023.
- FISCHETTI, M.; POLO, C.; SCANTAMBURLO, M. **A local branching heuristic for mixed-integer programs with 2-level variables, with an application to a telecommunication network design problem**. *Networks*, v. 44, p. 61-71, 2004. Disponível em: [doi:10.1002/net.20017](https://doi.org/10.1002/net.20017).
- FLEISCHMANN, B. **The discrete lot-sizing and scheduling problem**. *European Journal of Operational Research*, v. 44, n. 3, p. 337-348, 1990. Disponível em: [doi:10.1016/0377-2217\(90\)90245-7](https://doi.org/10.1016/0377-2217(90)90245-7).

- HAASE, K. **Capacitated lot-sizing with sequence dependent setup costs**. OR Spektrum, v. 18, p. 51-59, 1996. Disponível em: <https://doi.org/10.1007/BF01539882>.
- JAMES, R. J. W.; ALMADA-LOBO, B. **Single and parallel machine capacitated lotsizing and scheduling: new iterative MIP-based neighborhood search heuristics**. Computers & Operations Research, v. 38, n. 12, p. 1816-1825, 2011. Disponível em: <https://doi.org/10.1016/j.cor.2011.02.005>.
- KARMAKAR, U. S.; SCHRAGE, L. **The deterministic dynamic product cycling problem**. Operations Research, v. 33, n. 2, p. 326-345, 1985. Disponível em: doi:10.1287/opre.33.2.326.
- KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. **Optimization by simulated annealing**. Science, v. 220, n. 4598, p. 671-680, 1983. Disponível em: doi:10.1126/science.220.4598.671.
- MEDEIROS, M. A.; SOLER, W. A. O.; LOPES DE QUEIROZ, A. A. F. S. **Um modelo matemático para um problema de planejamento de produção em indústrias alimentícias**. 2022. Disponível em: <https://repositorio.ufms.br/handle/123456789/5105>.
- OLIVEIRA SOLER, W. A. de; POLDI, K. C.; SANTOS, M. O. **Capacitated lot sizing and scheduling with order acceptance and delivery time windows: mathematical model and a MIP-based heuristic**. SOBRAPO, 2019.
- SANTOS, H. G.; TOFFOLO, T. Â. M. **Tutorial de desenvolvimento de métodos de programação linear inteira mista em Python usando o pacote Python-MIP**. Pesquisa Operacional para o Desenvolvimento, v. 11, n. 3, p. 127-138, 2019. Disponível em: <https://www.podesenvolvimento.org.br/podesenvolvimento/article/view/629>.
- SERESHTI, N.; BIJARI, M. **Profit maximization in simultaneous lot-sizing and scheduling problem**. Applied Mathematical Modelling, v. 37, n. 23, p. 9516-9523, 2013. Disponível em: <https://doi.org/10.1016/j.apm.2013.05.004>.
- SOLER, W. A.; SANTOS, M. O. **A new branching rule to solve the capacitated lot sizing and scheduling problem with sequence dependent setups**. TEMA, 2017.
- TRIGEIRO, W. W.; THOMAS, L. J.; MCCLAIN, J. O. **Capacitated lot sizing with setup times**. Management Science, v. 35, n. 3, p. 353-366, 1989. Disponível em: <http://www.jstor.org/stable/2631977>.
- WAGNER, H. M.; WHITIN, T. M. **Dynamic version of the economic lot size model**. Management Science, v. 5, n. 1, p. 89-96, 1958. Disponível em: <http://www.jstor.org/stable/2626974>.

ANEXO

ANEXO 1 - FIGURA 1

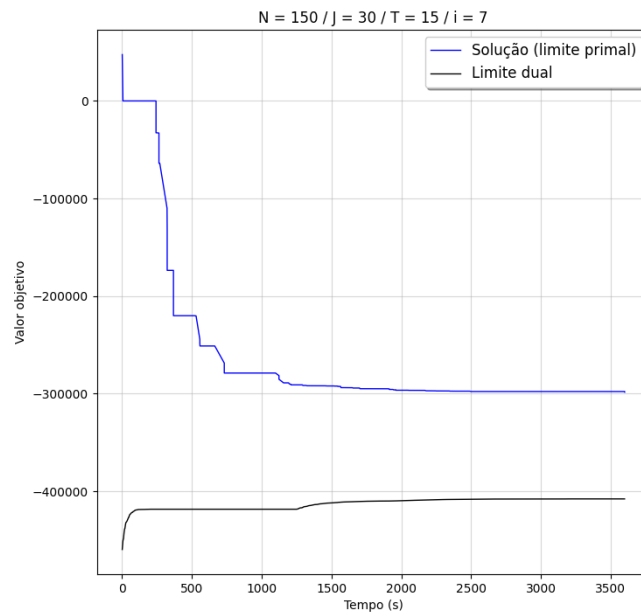
Figura 3 - Esquema do procedimento "RF" considerando $T = 5$, $windowSize = 2$, $overlap = 0.5$, e $step = (1 - overlap) * windowSize = 1$. Fonte: Soler, Poldi e Santos (2019)



APÊNDICES

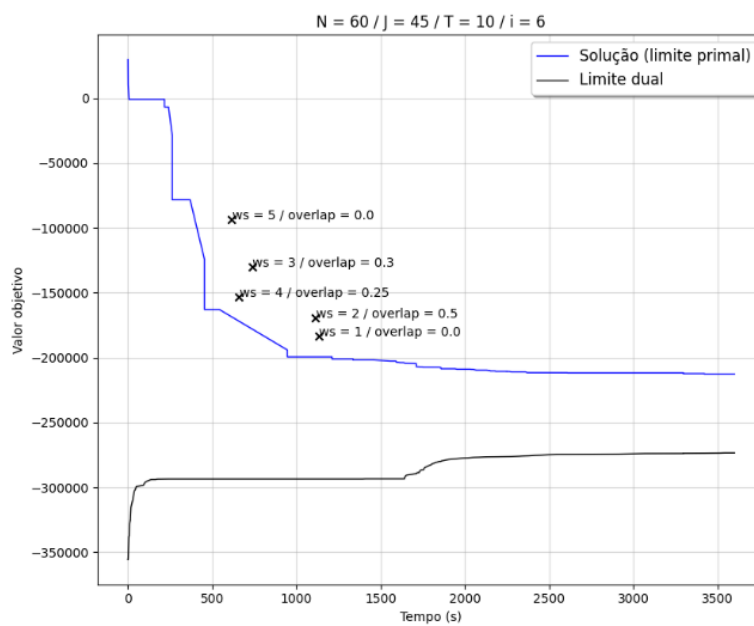
APÊNDICE 1 – FIGURA 2

Figura 2 – Exemplo de execução do algoritmo *Branch-and-Bound*. Fonte: Autor



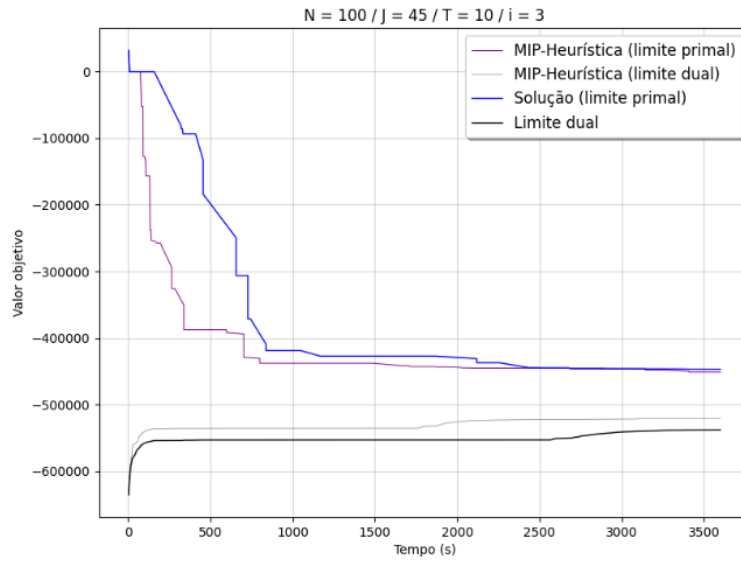
APÊNDICE 2 – FIGURA 3

Figura 3 – Análise da execução do RF em diversos parâmetros para uma instância. Fonte: Autor



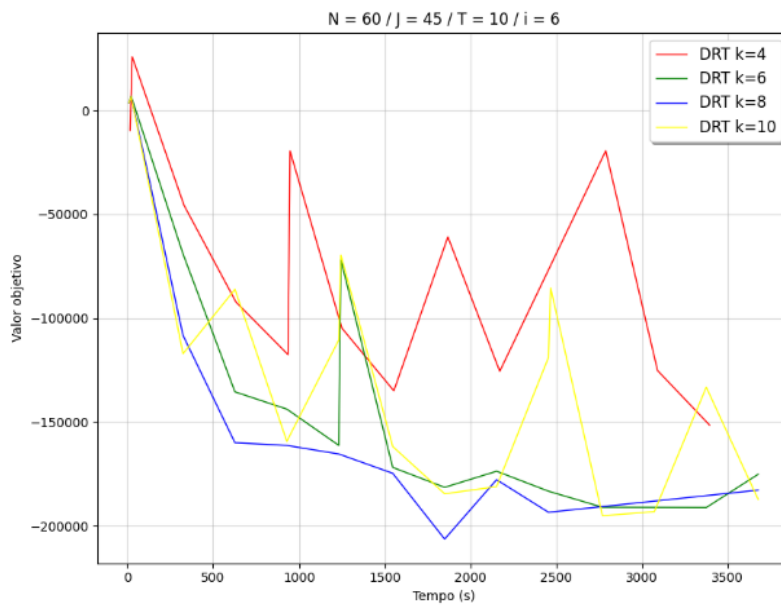
APÊNDICE 3 – FIGURA 4

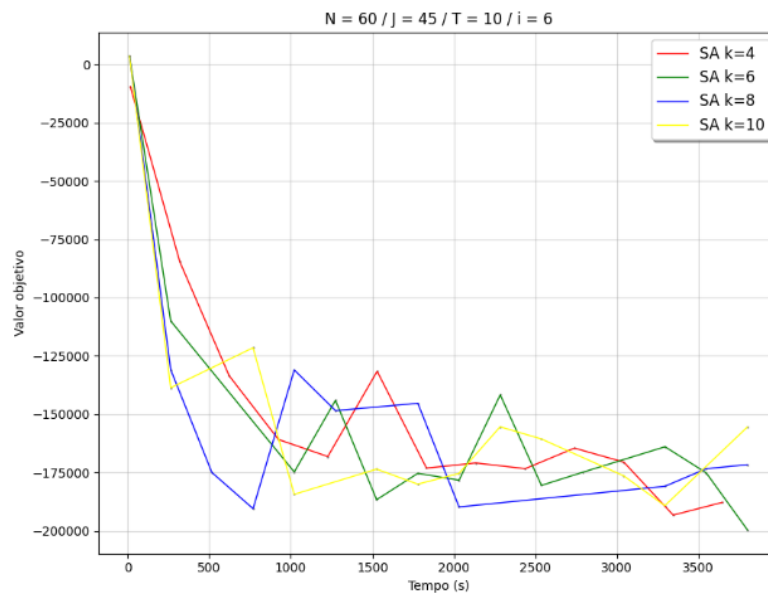
Figura 4 – Exemplo da execução da MIP-Heurística comparada ao *solver*. Fonte: Autor



APÊNDICE 4 – FIGURA 5

Figura 5 – Comparativo. Fonte: Autor





APÊNDICE 5 – TABELA 2

Tabela 2 – Gap médio entre os métodos. Fonte: Autor

Classe	Solver Gurobi 9.1.1	<i>Diversify, Refine & Tight-refine (DRT)</i>				<i>Simulated Annealing (SA)</i>				MIP-Heurística
		k1=4	k1=6	k1=8	k1=10	k=4	k=6	k=8	k=10	
N30 J15 T5	0,89	1,85	1,87	1,01	1,04	4,86	1,73	1,68	1,42	2,15
N30 J30 T5	12,88	12,42	11,75	11,95	13,09	14,19	13,14	13,35	13,53	10,55
N30 J45 T5	20,09	19,82	19,99	20,26	22,05	19,88	21,25	23,46	25,96	17,90
N50 J15 T5	3,47	3,76	3,41	3,68	3,30	8,49	3,77	3,73	3,51	3,77
N50 J30 T5	12,29	11,90	10,86	11,43	11,91	13,58	11,57	12,14	11,90	11,33
N50 J45 T5	15,74	15,94	15,08	15,59	15,60	15,73	16,04	16,37	17,12	15,02
N60 J45 T10	30,57	123,72	41,12	37,72	40,94	58,00	44,82	44,66	42,49	30,32
N100 J30 T10	20,78	42,66	30,92	25,38	26,87	34,92	25,27	25,39	22,93	19,51
N100 J45 T10	22,76	122,99	43,68	34,91	33,15	58,22	36,61	39,07	34,57	22,09
N150 J30 T15	32,67	117,47	77,03	57,70	53,38	73,57	63,41	51,95	42,14	32,58
Média	17,21	47,25	25,57	21,96	22,13	30,14	23,76	23,18	21,56	16,52

APÊNDICE 6 – TABELA 3

Tabela 3 – Desempenho do método Relax-and-Fix. Fonte: Autor

<i>Classe</i>	<i>Window size = 1 overlap = 0</i>	<i>Window size = 2 overlap = 0.5</i>	<i>Window size = 3 overlap = 0.3</i>	<i>Window size = 4 overlap = 0.25</i>	<i>Window size = 5 overlap = 0</i>
N30 J15 T5	14.83%	6.60%	4.34%	4.39%	3.97%
N30 J30 T5	19.36%	17.97%	18.89%	20.34%	17.16%
N30 J45 T5	25.93%	26.55%	23.71%	27.38%	24.63%
N50 J15 T5	9.55%	7.22%	6.96%	7.05%	6.41%
N50 J30 T5	14.35%	15.33%	14.98%	16.00%	14.79%
N50 J45 T5	17.46%	23.28%	18.22%	19.54%	18.13%
N60 J45 T10	46.64%	61.16%	124.79%	207.66%	180.77%
N100 J30 T10	21.99%	26.12%	42.68%	36.98%	35.62%
N100 J45 T10	32.69%	40.36%	70.44%	173.68%	134.45%
N150 J30 T15	66.17%	79.34%	83.97%	92.73%	131.33%
<i>Média</i>	<i>26.90%</i>	<i>30.39%</i>	<i>40.90%</i>	<i>60.58%</i>	<i>56.72%</i>

APÊNDICE 7 – TABELA 5

Tabela 5 – Análise comparativa dos métodos estudados, a partir da solução inicial fornecida pelo RF. Fonte: Autor.

Classe	Solver	DRT (k=8)	SA (k=10)
N30 J15 T5	0.91%	1.43%	1.28%
N30 J30 T5	12.23%	12.49%	14.07%
N30 J45 T5	19.51%	21.65%	22.50%
N50 J15 T5	3.34%	3.07%	3.29%
N50 J30 T5	11.17%	10.91%	11.57%
N50 J45 T5	14.66%	14.89%	15.24%
N60 J45 T10	29.45%	39.43%	41.53%
N100 J30 T10	18.62%	20.84%	21.20%
N100 J45 T10	24.10%	30.43%	31.56%
N150 J30 T15	30.85%	42.54%	43.43%
<i>Média</i>	<i>16.49%</i>	<i>19.77%</i>	<i>20.57%</i>