

# Aprimoramento de Metaheurísticas para Otimização da Cadeia de Suprimentos de Pistache

Raissa Gonçalves Diniz

Operations Research and Complex Systems Laboratory  
Universidade Federal de Minas Gerais  
Belo Horizonte, Brasil  
raissagd@ufmg.br

André Costa Batista

Departamento de Engenharia Elétrica  
Universidade Federal de Minas Gerais  
Belo Horizonte, Brasil  
andre-costa@ufmg.br

**Resumo**—As cadeias de suprimentos desempenham um papel fundamental na indústria agrícola, especialmente no caso de culturas de alto valor, como o pistache, onde uma gestão eficiente pode impulsionar o crescimento econômico e promover a sustentabilidade. Embora pesquisas anteriores tenham explorado estratégias de otimização para essas cadeias, a maioria dos estudos concentra-se no uso de algoritmos populacionais para enfrentar esses desafios. Neste trabalho, apresenta-se a aplicação do algoritmo de Busca em Vizinhança Variável (*Variable Neighborhood Search* - VNS) e avaliamos seu desempenho em comparação com uma metaheurística populacional, especificamente o Algoritmo Genético (*Genetic Algorithm* - GA). Os resultados mostram que o VNS apresenta consistentemente um desempenho superior ao GA, tanto em termos de eficiência computacional quanto de qualidade das soluções, especialmente em instâncias de maior porte. Esses achados destacam o potencial do método proposto como uma abordagem eficaz para a otimização de cadeias de suprimentos.

**Index Terms**—Otimização de Cadeias de Suprimentos, Metaheurísticas, Busca em Vizinhança Variável, Sustentabilidade.

## I. INTRODUÇÃO

A agricultura desempenha um papel fundamental no desenvolvimento econômico, contribuindo não apenas para a segurança alimentar, mas também para as exportações e o crescimento industrial. Entre as culturas de alto valor, os pistaches se destacam no comércio internacional, com os Estados Unidos, a Turquia e o Irã liderando a produção global [22]. A gestão eficiente de cadeias de suprimentos é essencial para manter a qualidade dos produtos, reduzir impactos ambientais e garantir uma distribuição eficaz. Um dos principais desafios na produção de pistache é o manejo dos subprodutos, como cascas e resíduos da polpa, que, se descartados de forma inadequada, podem comprometer a qualidade do solo e gerar riscos à saúde devido à contaminação por aflatoxinas produzidas por fungos [26]. Estratégias sustentáveis de gestão de resíduos, como compostagem e produção de biogás, têm sido exploradas para mitigar esses problemas, conforme discutido por [6].

Falhas na cadeia de suprimentos exercem impacto significativo tanto na qualidade dos produtos quanto nas operações comerciais. Estudos anteriores, como o de [2], investigaram técnicas de otimização baseadas em Algoritmos Genéticos (*Genetic Algorithm* - GA) e outras meta-heurísticas evolucionárias para a logística do pistache. No entanto, muitas

dessas comparações se baseiam apenas nas melhores soluções encontradas, sem uma validação estatística robusta que comprove a superioridade dos resultados. Além disso, ainda há debate sobre a influência real das diferenças entre meta-heurísticas bioinspiradas na qualidade das soluções obtidas.

Neste estudo, propõe-se a otimização da cadeia de suprimentos do pistache utilizando a metaheurística Busca em Vizinhança Variável (*Variable Neighborhood Search* - VNS), que é amplamente reconhecida por sua eficácia na resolução de problemas combinatórios complexos com baixa necessidade de parametrização. O modelo matemático utilizado é baseado na formulação mista linear proposta por [2] para a logística direta e reversa, que foca na minimização de custos e na redução de impactos ambientais por meio do reaproveitamento de subprodutos. A principal contribuição deste trabalho está na aplicação do VNS a esse modelo, avaliando seu desempenho em comparação com o Algoritmo Genético (GA). Os resultados demonstram que o VNS apresenta vantagens em termos de qualidade das soluções e tempo computacional. Embora o uso de metaheurísticas para otimização de cadeias de suprimentos não seja novo, a aplicação específica do VNS neste contexto oferece uma alternativa interessante e promissora, especialmente em problemas de maior escala e complexidade. Essa abordagem pode contribuir para o aprimoramento da eficiência na gestão de cadeias de suprimentos agrícolas, mantendo a sustentabilidade e otimizando os custos operacionais.

A estrutura deste artigo organiza-se da seguinte forma: a Seção II apresenta a definição do problema, descrevendo a estrutura da cadeia de suprimentos e os objetivos de otimização. A Seção III detalha a abordagem heurística, explicando a metodologia VNS e sua implementação. A Seção IV compara os resultados experimentais entre VNS e GA. Por fim, a Seção V apresenta as principais conclusões e direções para pesquisas futuras.

## II. DEFINIÇÃO DO PROBLEMA

A rede logística proposta por [2], ilustrada na Figura 1, descreve o fluxo de produtos e subprodutos ao longo da cadeia de suprimentos do pistache. O processo tem início com os produtores ( $i$ ), que enviam os pistaches colhidos aos centros de processamento ( $j$ ). Nesses centros, os pistaches são

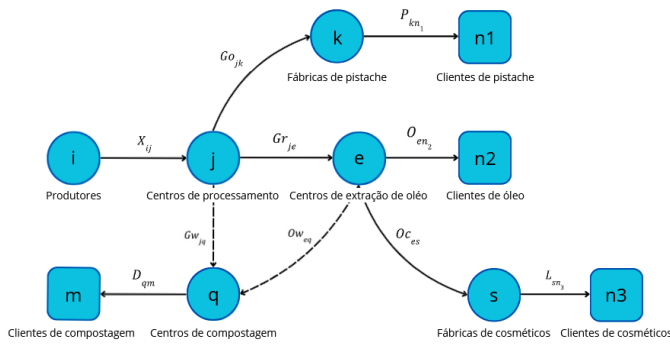


Figura 1: Estrutura da cadeia de suprimentos do pistache proposta.

transformados em três componentes principais: pistaches de casca aberta, sementes cruas e resíduos orgânicos.

Cada um desses componentes segue fluxos específicos. Os pistaches de casca aberta são encaminhados às fábricas de pistache ( $k$ ), que os processam e os distribuem aos clientes finais ( $n1$ ). As sementes cruas são enviadas aos centros de extração de óleo ( $e$ ), responsáveis pela produção de óleo de pistache, o qual é direcionado tanto a clientes finais ( $n2$ ) quanto às fábricas de cosméticos ( $s$ ). Estas, por sua vez, processam o óleo em produtos cosméticos que são fornecidos aos respectivos consumidores ( $n3$ ). Os resíduos orgânicos gerados no processamento inicial são destinados aos centros de compostagem ( $q$ ), onde são transformados em composto, posteriormente distribuído aos clientes de compostagem ( $m$ ).

Embora, em termos práticos, o composto produzido possa ser reutilizado pelos próprios produtores agrícolas, estabelecendo um sistema de circuito fechado, o modelo matemático adotado não representa explicitamente essa retroalimentação. Em vez disso, os clientes de composto não estão diretamente associados aos produtores originais, caracterizando um ciclo de feedback implícito. Essa abordagem, já adotada em trabalhos anteriores ([11], [19], [20]), preserva a simplicidade estrutural do modelo. No entanto, a inclusão de uma modelagem rigorosa desse circuito de retorno representa uma direção promissora para investigações futuras, especialmente no contexto de cadeias de suprimentos sustentáveis.

O problema linear misto visa minimizar os custos totais (1), incluindo custos de abertura, produção e transporte, sob as seguintes restrições de capacidade e demanda, não tolerando escassez (ver mais detalhes sobre as variáveis na Tabela 5 de [2]):

$$\min z_1 + z_2 + z_3 \quad (1)$$

$$\text{s.t.:} \quad \sum_{j=1}^J X_{ij} \leq Cpa_i, \quad \forall i \in I \quad (2)$$

$$\sum_{i=1}^I X_{ij} \leq Cpu_j \times U_j, \quad \forall j \in J \quad (3)$$

$$\sum_{e=1}^E Ow_{eq} + \sum_{j=1}^J Gw_{jq} \leq Cpy_q \times Y_q, \quad \forall q \in Q \quad (4)$$

$$\sum_{j=1}^J Go_{jk} \leq Cpw_k \times W_k, \quad \forall k \in K \quad (5)$$

$$\sum_{j=1}^J Gr_{je} \leq Cpr_e \times R_e, \quad \forall e \in E \quad (6)$$

$$\sum_{e=1}^E O_{en_2} \leq Cpv_s \times V_s, \quad \forall s \in S \quad (7)$$

$$\sum_{k=1}^K Go_{jk} + \sum_{e=1}^E Gr_{je} + \sum_{q=1}^Q Gw_{jq} \leq \sum_{i=1}^I X_{ij}, \quad \forall j \in J \quad (8)$$

$$\sum_{k=1}^K Go_{jk} = (1 - \beta) \sum_{i=1}^I X_{ij} \theta_1, \quad \forall j \in J \quad (9)$$

$$\sum_{e=1}^E Gr_{je} = (1 - \beta) \sum_{i=1}^I X_{ij} \theta_2, \quad \forall j \in J \quad (10)$$

$$\sum_{q=1}^Q Gw_{jq} = \sum_{i=1}^I X_{ij} \theta_3, \quad \forall j \in J \quad (11)$$

$$\theta_1 + \theta_2 + \theta_3 = 1 \quad (12)$$

$$\sum_{n_1=1}^{N_1} P_{kn_1} \leq \sum_{j=1}^J Go_{jk} \times \gamma_k, \quad \forall k \in K \quad (13)$$

$$\sum_{n_2=1}^{N_2} O_{en_2} + \sum_{s=1}^S Oc_{es} \leq \sum_{j=1}^J Gr_{je} \times (1 - \lambda), \quad \forall e \in E \quad (14)$$

$$\sum_{q=1}^Q Ow_{eq} \leq \sum_{j=1}^J Gr_{je} \times \lambda, \quad \forall e \in E \quad (15)$$

$$\sum_{n_3=1}^{N_3} L_{sn_3} \leq \sum_{e=1}^E Oc_{es} \times \gamma_s, \quad \forall s \in S \quad (16)$$

$$\sum_{m=1}^M D_{qm} \leq \left( \sum_{j=1}^J Gw_{jq} + \sum_{e=1}^E Ow_{eq} \right) \times \gamma_q, \quad \forall q \in Q \quad (17)$$

$$\sum_{k=1}^K P_{kn_1} \geq Dp_{n_1}, \quad \forall n_1 \in N_1 \quad (18)$$

$$\sum_{e=1}^E O_{en_2} \geq Du_{n_2}, \quad \forall n_2 \in N_2 \quad (19)$$

$$\sum_{s=1}^S L_{sn_3} \geq Ds_{n_3}, \quad \forall n_3 \in N_3 \quad (20)$$

$$\sum_{q=1}^Q D_{qm} \leq Dc_m, \quad \forall m \in M \quad (21)$$

- A função objetivo (1) minimiza os custos totais da cadeia de suprimentos, incluindo custos de abertura, transporte e

produção. Os custos são divididos em três componentes:  $z_1$  (22),  $z_2$  (23), e  $z_3$  (24), que representam os custos de abertura de instalações, transporte e produção, respectivamente, conforme mostrado nas seguintes equações:

$$z_1 = \sum_{j=1}^J Fu_j \cdot U_j + \sum_{q=1}^Q Fy_q \cdot Y_q + \sum_{k=1}^K Fw_k \cdot W_k + \sum_{e=1}^E Fr_e \cdot R_e + \sum_{s=1}^S Fv_s \cdot V_s \quad (22)$$

$$z_2 = \sum_{i=1}^I \sum_{j=1}^J CI_i \cdot X_{ij} + \sum_{j=1}^J \sum_{k=1}^K Cu_{1j} \cdot Go_{jk} + \sum_{j=1}^J \sum_{e=1}^E Cu_{2j} \cdot Gr_{je} + \sum_{q=1}^Q \sum_{m=1}^M Cy_q \cdot D_{qm} + \sum_{k=1}^K \sum_{n_1=1}^{N_1} Cw_k \cdot P_{kn_1} + \sum_{e=1}^E Cr_e \cdot \left( \sum_{n_2=1}^{N_2} O_{en_2} + \sum_{s=1}^S Oc_{es} \right) + \sum_{s=1}^S \sum_{n_3=1}^{N_3} Cv_s \cdot L_{sn_3} \quad (23)$$

$$z_3 = \sum_{i=1}^I \sum_{j=1}^J CX_{ij} \cdot X_{ij} + \sum_{j=1}^J \sum_{k=1}^K CK_{jk} \cdot Go_{jk} + \sum_{j=1}^J \sum_{q=1}^Q CJ_{jq} \cdot Gw_{jq} + \sum_{e=1}^E \sum_{s=1}^S CS_{es} \cdot Oc_{es} + \sum_{e=1}^E \sum_{n_2=1}^{N_2} CN_{en_2} \cdot Oc_{en_2} + \sum_{e=1}^E \sum_{q=1}^Q CQ_{eq} \cdot Ow_{eq} + \sum_{s=1}^S \sum_{n_3=1}^{N_3} Cl_{sn_3} \cdot L_{sn_3} + \sum_{k=1}^K \sum_{n_1=1}^{N_1} Cp_{kn_1} \cdot P_{kn_1} + \sum_{q=1}^Q \sum_{m=1}^M Cd_{qm} \cdot D_{qm} \quad (24)$$

- As variáveis  $R_j$ ,  $Y_q$ ,  $W_k$ ,  $Z_e$  e  $V_s$  são binárias e determinam se certos elementos do modelo estão ativos. Por exemplo,  $R_j = 1$  indica que a instalação  $j$  está operacional, e  $R_j = 0$  caso contrário. As demais variáveis do modelo são contínuas e não negativas, representando valores reais maiores ou iguais a zero. Essas variáveis modelam fluxos, quantidades e outros fatores mensuráveis dentro do sistema.
- As restrições (2)-(7) tratam das limitações de capacidade e demanda de cada instalação na cadeia de suprimentos. Em particular, a restrição (2) limita a quantidade de produtos transportados dos produtores para os centros de distribuição. Já as restrições (3)-(7) garantem que as

quantidades enviadas para os centros de processamento, compostagem, fábricas de pistache, centros de extração de óleo e fábricas de cosméticos não ultrapassem as capacidades dessas instalações.

- A quantidade de produtos enviados de qualquer instalação não deve ultrapassar os pedidos recebidos por essa instalação. Consequentemente, a restrição (8) estabelece que a quantidade de produtos expedidos de um centro de processamento deve ser menor ou igual à quantidade recebida por outras instalações. As restrições (9)-(12) definem as quantidades de pistaches de casca aberta, sementes cruas e resíduos gerados durante o processo de descascamento em cada centro de processamento. De forma similar, as restrições (13)-(17) regem os fluxos dentro das fábricas de pistache, centros de extração de óleo, fábricas de cosméticos e centros de compostagem.
- Por fim, para garantir que a demanda dos clientes seja atendida, as restrições (18)-(21) asseguram que as quantidades de produtos entregues aos clientes de pistache, óleo de pistache, cosméticos e composto satisfaçam ou superem suas respectivas demandas.

### III. METODOLOGIA

Esta seção apresenta a metodologia utilizada para resolver o problema de otimização, empregando codificação baseada em prioridade [12] e o algoritmo VNS. A codificação baseada em prioridade ordena as variáveis de decisão para explorar o espaço de soluções mantendo a viabilidade, enquanto o VNS alterna entre diferentes estruturas de vizinhança para escapar de ótimos locais. A abordagem proposta inclui operadores de vizinhança personalizados e configurações de parâmetros adequadas.

#### A. Representação da Solução

Seguindo [2] e [12], é utilizada a codificação baseada em prioridade para representar as soluções. Cada par origem-destino é tratado como um gene, cujas prioridades orientam a construção da árvore de transporte. O processo de decodificação seleciona iterativamente o nó de maior prioridade, conectando-o à alternativa de menor custo, garantindo sempre o cumprimento das restrições de viabilidade.

Diferentemente de [2], que utiliza codificação real com valores contínuos entre 0 e 1, emprega-se uma abordagem baseada em números inteiros sequenciais de 1 a  $Z$ , onde  $Z$  representa o número total de elementos considerados na ordenação de prioridades para um dado fluxo logístico (por exemplo, a soma entre fontes e destinos de transporte em determinado segmento da cadeia). Essa representação ordinal, atribuída com base nos custos médios de transporte, visa aprimorar a interpretabilidade das soluções e facilitar a implementação de operadores de vizinhança. Além disso, a codificação inteira permite uma distinção mais clara entre soluções vizinhas, o que se mostra vantajoso em problemas combinatórios, como é o caso da otimização de cadeias de suprimentos. A eficácia desse tipo de codificação já foi demonstrada em estudos anteriores ([11]–[13]).

A Figura 2 apresenta a estrutura do cromossomo adotado, composto por oito segmentos ( $S_1$  a  $S_8$ ), cada um responsável por ordenar as prioridades em um fluxo logístico específico da cadeia de suprimentos. A parte inferior da figura ilustra um exemplo de cromossomo completo, com prioridades inteiras associadas a cada par origem-destino. Essa estrutura modular facilita a aplicação de operadores de vizinhança e o processo de decodificação baseado em transporte prioritário.

| S1 | S2 | S3 | S4 | S5 | S6   | S7  | S8 |
|----|----|----|----|----|------|-----|----|
| K  | N1 | S  | N3 | E  | N2+S | J   | K  |
| J  | E  | I  | J  | Q  | M    | J+E | Q  |

| S1    | S2    | S3  | S4 | S5    | S6      | S7        | S8    |
|-------|-------|-----|----|-------|---------|-----------|-------|
| 1 4 5 | 3 2   | 3 2 | 1  | 1 2 3 | 4 5 6   | 1         | 2 3 4 |
| 4 3   | 1 2 5 | 2 3 | 1  | 3 6   | 4 2 5 1 | 4 2 5 6 7 | 1 3   |

Figura 2: Representação da estrutura do cromossomo baseado em prioridade. A parte inferior mostra um exemplo de um cromossomo preenchido com as prioridades inteiras.

O processo de decodificação garante a conformidade com as restrições (1)-(21), determinando sequencialmente os fluxos dos produtores para os centros de processamento, fábricas e consumidores, minimizando os custos de transporte. Ajustes são feitos para priorizar as entidades com maior necessidade, assegurando a viabilidade. Da mesma forma, a compostagem e o gerenciamento de resíduos seguem restrições rigorosas de capacidade e demanda para otimizar a distribuição.

A Figura 3 ilustra o processo de decodificação, que inicia com o fluxo de produtos derivados do pistache das fábricas para os consumidores, garantindo o atendimento de todas as restrições. Os fluxos de óleo e produtos cosméticos seguem princípios semelhantes. Os centros de processamento regulam a alocação do pistache, considerando perdas produtivas e garantindo uma distribuição viável. Os pistaches colhidos são distribuídos proporcionalmente entre fábricas e centros de extração de óleo, priorizando o atendimento da demanda e minimizando custos de transporte. Os fluxos entre produtores e centros de processamento não podem exceder a capacidade de produção, garantindo a viabilidade do sistema. Os fluxos de resíduos e fertilizantes são gerenciados de forma semelhante, respeitando os limites de capacidade de processamento e compostagem, enquanto minimizam os custos.

Para a decodificação de cada fluxo logístico da cadeia de suprimentos, utiliza-se a função *decodingStep*, que implementa um algoritmo de transporte baseado em prioridades, conforme apresentado no *Procedure 1* de [12]. Essa função atua iterativamente, alocando unidades de produto de uma fonte para um destino com base em um vetor cromossômico que define a ordem de prioridade entre os nós logísticos, sempre respeitando as capacidades das fontes e as demandas dos destinos, além dos custos unitários de transporte. Os Algoritmos 1 e 2 detalham o processo completo de decodificação de uma solução. O Algoritmo 1 inicia o processo ao decodificar os fluxos principais entre fábricas, centros de

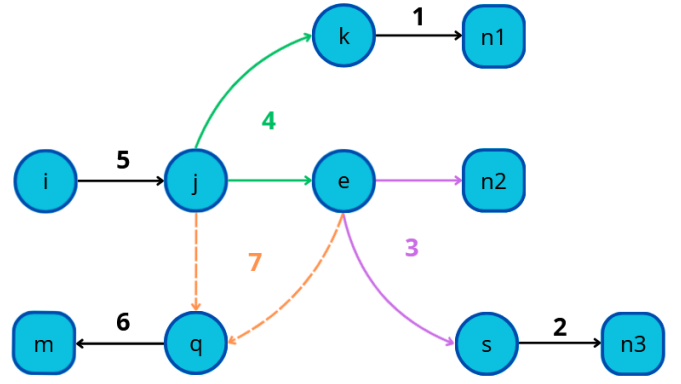


Figura 3: Etapas de decodificação do fluxo na cadeia de suprimentos

extração, consumidores e centros de processamento, utilizando os segmentos cromossômicos  $S_1$  a  $S_5$ . Já o Algoritmo 2 realiza ajustes adicionais nos fluxos para garantir a coerência entre entradas e saídas nos centros intermediários, decodifica o transporte dos produtores para os centros de processamento ( $S_6$ ), aloca os resíduos gerados nos processos para centros de compostagem ( $S_7$  e  $S_8$ ), e determina quais instalações devem ser efetivamente abertas com base nos fluxos obtidos. Ao final do processo, obtém-se um conjunto completo de variáveis de decisão contínuas e binárias que representam uma solução viável e coerente para o problema de otimização proposto.

#### B. Busca em Vizinhança Variável (VNS)

O VNS explora o espaço de soluções alterando sistematicamente as estruturas de vizinhança [27]. O algoritmo alterna entre as fases de perturbação (*shaking*) e busca local, aceitando novas soluções apenas se melhorarem a função objetivo. Essa abordagem tem se mostrado eficaz na resolução de problemas de otimização com restrições ([9], [14], [15]).

A implementação do algoritmo VNS utilizado neste estudo está descrita no Algoritmo 3.

#### C. Estruturas de Vizinhança

Quatro operadores clássicos de vizinhança [7], ilustrados na Figura 4, são utilizados para aumentar a diversidade das soluções:

- **Swap:** Permuta dois elementos no cromossomo.
- **Reversion:** Inverte a sequência de elementos entre dois índices selecionados [17].
- **Insertion:** Move um elemento para uma nova posição, deslocando os demais conforme necessário [16].
- **Slide:** Remove um elemento, desloca os outros e o reinsere em uma nova posição.

Cada operação é aplicada a um vetor de prioridade selecionado aleatoriamente dentro do conjunto de soluções, garantindo a viabilidade e aumentando a eficiência da busca.

---

**Algorithm 1** Decodificação do cromossomo para a cadeia de suprimentos (Parte 1)

---

```
1: procedure DECODE(dados)
2:   Entrada: dados estruturados contendo:
3:   - Capacidades e demandas para todas as instalações
4:   - Custos unitários de transporte
5:   - Taxas de perda para cada estágio do processo
6:   - Demandas dos consumidores
7:   - Segmentos de cromossomo ( $S_1, S_2, \dots, S_8$ ) representando prioridades de transporte
8:   Saída: custototal, variáveis de decisão contínuas ( $X, P, L, O, O_c, G_o, G_r, D, G_w, O_w$ ) e variáveis binárias de abertura ( $U, W, R, V, Y$ )
9:   // Passo 1: Inicializar variáveis e ler dados de entrada
10:  custototal  $\leftarrow$  0
11:  // Passo 2: Decodificar Fábricas de Pistache  $\rightarrow$  Consumidores de Pistache
12:   $a_1 \leftarrow$  capacidades das fábricas de pistache
13:   $b_1 \leftarrow$  demandas dos consumidores de pistache
14:   $c_1 \leftarrow$  custos de transporte
15:  ( $custo1, P$ )  $\leftarrow$  decodingStep( $S_1, a_1, b_1, c_1$ )
16:  custototal  $\leftarrow$  custototal + custo1
17:  // Passo 3: Decodificar Fábricas de Cosméticos  $\rightarrow$  Consumidores de Cosméticos
18:   $a_2 \leftarrow$  capacidades das fábricas de cosméticos
19:   $b_2 \leftarrow$  demandas dos consumidores de cosméticos
20:   $c_2 \leftarrow$  custos de transporte
21:  ( $custo2, L$ )  $\leftarrow$  decodingStep( $S_2, a_2, b_2, c_2$ )
22:  custototal  $\leftarrow$  custototal + custo2
23:  // Passo 4: Decodificar Centros de Extração de Óleo  $\rightarrow$  (Consumidores de Óleo + Fábricas de Cosméticos)
24:   $a_3 \leftarrow$  capacidades dos centros de extração de óleo
25:   $b_3 \leftarrow$  demandas dos consumidores de óleo e fábricas de cosméticos
26:   $c_3 \leftarrow$  custos de transporte
27:  ( $custo3, OO_c$ )  $\leftarrow$  decodingStep( $S_3, a_3, b_3, c_3$ )
28:  Separar  $OO_c$  em  $O$  (consumidores de óleo) e  $O_c$  (fábricas de cosméticos)
29:  custototal  $\leftarrow$  custototal + custo3
30:  // Passo 5: Decodificar Centro de Processamento  $\rightarrow$  Fábricas de Pistache
31:   $a_4 \leftarrow$  capacidades dos centros de processamento
32:   $b_4 \leftarrow$  demandas ajustadas das fábricas de pistache
33:   $c_4 \leftarrow$  custos de transporte
34:  ( $\_, G_o$ )  $\leftarrow$  decodingStep( $S_4, a_4, b_4, c_4$ )
35:  // Passo 6: Decodificar Centro de Processamento  $\rightarrow$  Centros de Extração de Óleo
36:   $a_5 \leftarrow$  capacidades dos centros de processamento
37:   $b_5 \leftarrow$  demandas ajustadas dos centros de extração de óleo
38:   $c_5 \leftarrow$  custos de transporte
39:  ( $\_, G_r$ )  $\leftarrow$  decodingStep( $S_5, a_5, b_5, c_5$ )
40: end procedure
```

---

---

**Algorithm 2** Decodificação do cromossomo para a cadeia de suprimentos (Parte 2)

---

```
1: procedure DECODE(continuação)
2:   // Passo 7: Ajustar demandas nos Centros de Processamento
3:   for cada centro de processamento  $j$  do
4:     Calcular saídas necessárias com base nos fluxos de pistache e óleo
5:     Atualizar  $G_o$  e  $G_r$  para corresponder às capacidades efetivas
6:   end for
7:   // Passo 8: Decodificar Produtores de Pistache  $\rightarrow$  Centros de Processamento
8:    $a_6 \leftarrow$  capacidades dos produtores de pistache
9:    $b_6 \leftarrow$  demandas atualizadas nos centros de processamento
10:   $c_6 \leftarrow$  custos de transporte
11:  ( $custo6, X$ )  $\leftarrow$  decodingStep( $S_6, a_6, b_6, c_6$ )
12:  custototal  $\leftarrow$  custototal + custo6
13:  // Passo 9: Decodificar Centros de Compostagem  $\rightarrow$  Consumidores de Compostagem
14:   $a_7 \leftarrow$  capacidades dos centros de compostagem
15:   $b_7 \leftarrow$  demandas dos consumidores de compostagem
16:   $c_7 \leftarrow$  custos de transporte
17:  ( $custo7, D$ )  $\leftarrow$  decodingStep( $S_7, a_7, b_7, c_7$ )
18:  custototal  $\leftarrow$  custototal + custo7
19:  // Passo 10: Atribuir fluxos de resíduos
20:  Calcular resíduos dos centros de processamento
21:   $a_8 \leftarrow$  capacidades disponíveis dos centros de compostagem
22:   $b_8 \leftarrow$  demandas de resíduos dos centros de processamento e extração de óleo
23:   $c_8 \leftarrow$  custos de transporte
24:  ( $\_, O_w$ )  $\leftarrow$  decodingStep( $S_8, a_8, b_8, c_8$ )
25:  Atualizar custototal
26:  // Passo 11: Definir variáveis binárias de abertura
27:   $U \leftarrow$  centros de processamento abertos
28:   $W \leftarrow$  fábricas de pistache abertas
29:   $R \leftarrow$  centros de extração de óleo abertos
30:   $V \leftarrow$  fábricas de cosméticos abertas
31:   $Y \leftarrow$  centros de compostagem abertas
32:  return custototal, variáveis de decisão e variáveis binárias de abertura
33: end procedure
```

---

#### IV. EXPERIMENTOS COMPUTACIONAIS

O método heurístico proposto foi testado em instâncias de diferentes tamanhos, definidos pelo número de elementos em cada entidade do problema ( $I, J, K, E, S, Q, N_1, N_2, N_3$  e  $M$ ). Os tamanhos testados foram 100, 200, 400, 800 e 1600, resultando em contagens totais de variáveis de 100.500; 401.000; 1.602.000; 6.404.000; e 25.608.000, respectivamente. Entretanto, algoritmos que utilizam codificação baseada em prioridade possuem um número significativamente menor de

---

**Algorithm 3** Busca em Vizinhança Variável (VNS)

---

**Require:** Solução inicial  $x$ , tamanho máximo da vizinhança

$k_{\max}$ , limite de tempo  $t_{\max}$

1: **Saída:** Melhor solução encontrada  $x^*$

2:  $x \leftarrow x$

3: **repeat**

4:    $k \leftarrow 1$

5:   **repeat**

6:      $x' \leftarrow \text{Shake}(x, k)$

7:      $x'' \leftarrow \text{LocalSearch}(x')$

8:     **if**  $f(x'') < f(x)$  **then**

9:        $x \leftarrow x''$ ,  $k \leftarrow 1$

10:    **else**

11:       $k \leftarrow k + 1$

12:    **end if**

13:   **until**  $k > k_{\max}$

14:    $t \leftarrow \text{CpuTime}()$

15: **until**  $t > t_{\max}$ 

---

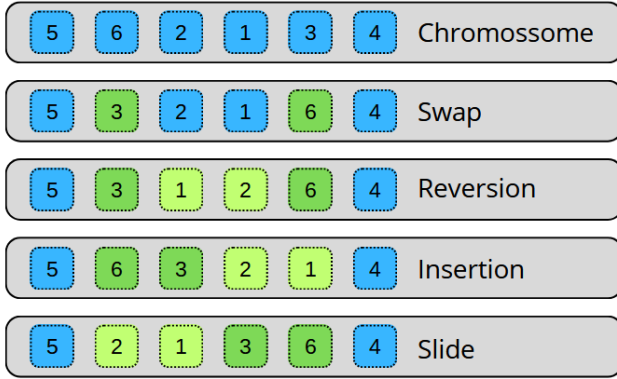


Figura 4: Estruturas de vizinhança utilizadas no algoritmo VNS

variáveis, pois consideram apenas as prioridades dos pares fonte-depósito. Nesses casos, as contagens de variáveis foram reduzidas para 1.700; 3.400; 6.800; 13.600; e 27.200.

Os parâmetros das instâncias, como capacidades e demandas, foram gerados aleatoriamente conforme a Tabela 7 de [2]. A abordagem baseada no VNS foi comparada com o GA, que foi uma das meta-heurísticas evolucionárias utilizadas na referência considerada. O GA utilizou mutações baseadas em troca (*swap-based mutations*) e cruzamentos de subsequências (*subsequence-exchanging crossovers*), com vetores selecionados aleatoriamente, uma população de 100 indivíduos e uma estratégia de seleção por Torneio Binário. Além disso, seus outros principais parâmetros foram definidos com base em valores tradicionais da literatura: taxa de cruzamento de 0,8 e taxa de mutação de 0,2. O processo de inicialização utilizou geração estocástica dos cromossomos.

Tanto o VNS quanto o GA foram executados 30 vezes por instância. O critério de parada adotado foi um número máximo de avaliações da função objetivo, definido empiricamente a

partir de testes sucessivos, de forma a equilibrar custo computacional e qualidade das soluções. Especificamente, foram utilizadas 6.250 avaliações para instâncias de tamanho 100; 12.500 para 200; 25.000 para 400; 50.000 para 800; e 100.000 para 1.600. Adicionalmente, o VNS também interrompe a execução caso todos os operadores de vizinhança falhem em melhorar a solução atual.

Os experimentos foram conduzidos em um sistema com Ubuntu 24.04.1 LTS, processador Intel Xeon E3-1240 V2 (8 núcleos) e 32 GB de RAM.

#### A. Comparação entre VNS e GA

A comparação com o GA segue a metodologia de [2], embora uma correspondência exata não tenha sido possível devido às diferenças no processo de decodificação. A Tabela I apresenta o desempenho médio das 30 melhores soluções obtidas por cada algoritmo. Para instâncias pequenas (tamanho 100), o VNS superou o GA com uma melhoria de 4,48%. À medida que o tamanho do problema aumentou, essa superioridade se manteve e, em alguns casos, foi ainda mais acentuada. O VNS apresentou melhorias variando de 4,48% a 9,87% em relação ao GA nas instâncias maiores, com destaque para o ganho de 9,87% na instância de 200 elementos.

Os boxplots na Fig. 5 ilustram a distribuição das soluções, com diferenças mais significativas nas instâncias maiores. Esses resultados sugerem que o VNS tem um desempenho superior ao GA em problemas de grande escala. Além disso, o GA apresentou tempos de execução muito mais longos, provavelmente devido ao critério de parada adicional do VNS, que leva em consideração o esgotamento dos operadores de vizinhança, enquanto o GA depende de um número fixo de avaliações da função objetivo. Ao mesmo tempo, isso sugere que o VNS pode estar sendo mais eficiente em encontrar boas soluções com um número menor de avaliações.

Os testes estatísticos apresentados na Tabela II reforçam essas observações: nas instâncias 800 e 1600, ambas as distribuições passaram no teste de normalidade de Shapiro-Wilk ( $p > 0.05$ ), permitindo o uso do teste t de Welch, que indicou diferenças estatisticamente significativas ( $p < 0,05$ ). Já nas demais instâncias, pelo menos uma das distribuições não seguiu normalidade, sendo aplicado o teste de Mann-Whitney, que também revelou diferenças significativas. Assim, conclui-se que o VNS foi consistentemente mais eficaz estatisticamente, especialmente à medida que o tamanho do problema aumentou.

Em resumo, embora o GA tenha um desempenho competitivo em instâncias pequenas, sua escalabilidade é limitada. Em contraste, o VNS não apenas obtém soluções superiores, mas também demonstra maior eficiência no tratamento de problemas de grande escala, estabelecendo-se como um método de otimização mais robusto e escalável.

## V. CONCLUSÃO

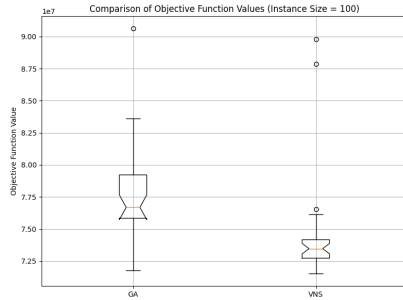
Este estudo apresenta um modelo de otimização para a cadeia de suprimentos de pistache, abordando as complexidades do gerenciamento de produtos e subprodutos ao longo de

Tabela I: Comparação entre o algoritmo VNS e uma meta-heurística baseada em população (GA) em termos do valor médio da função objetivo das melhores soluções e do tempo médio de execução.

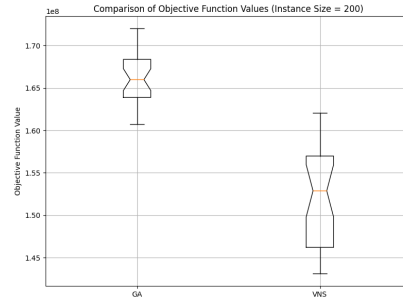
| Instância | Avaliação da função objetivo (média) |                    |               | Tempo de Execução (média) |            |               |
|-----------|--------------------------------------|--------------------|---------------|---------------------------|------------|---------------|
|           | VNS                                  | GA                 | Diferença (%) | VNS                       | GA         | Diferença (%) |
| 100       | $7.44 \times 10^7$                   | $7.78 \times 10^7$ | 4.48          | 369.59s                   | 60,480.38s | 197.57        |
| 200       | $1.51 \times 10^8$                   | $1.66 \times 10^8$ | 9.87          | 1,192.69s                 | 23,210.39s | 180.45        |
| 400       | $3.12 \times 10^8$                   | $3.42 \times 10^8$ | 9.17          | 3,917.46s                 | 27,590.31s | 150.27        |
| 800       | $6.58 \times 10^8$                   | $7.08 \times 10^8$ | 7.36          | 11,512.24s                | 48,803.09s | 123.65        |
| 1600      | $1.36 \times 10^9$                   | $1.43 \times 10^9$ | 4.64          | 17,767s                   | 217,958s   | 1126.95       |

Tabela II: Resultados dos testes estatísticos para as instâncias analisadas. Inclui o p-valor do teste de normalidade (Shapiro-Wilk) para o GA e o VNS, e o teste de comparação adequado (Mann-Whitney ou t-teste), com seus respectivos p-valores.

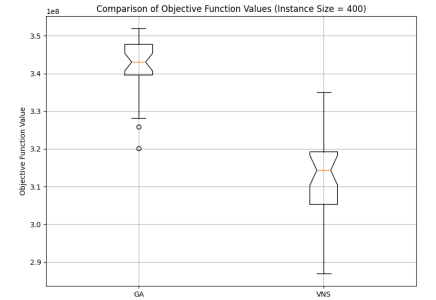
| Instância | Shapiro GA (p-valor) | Shapiro VNS (p-valor) | Teste Comparativo | p-valor do Teste       |
|-----------|----------------------|-----------------------|-------------------|------------------------|
| 100       | 0.00039              | $1.77 \times 10^{-8}$ | Mann-Whitney      | $2.80 \times 10^{-7}$  |
| 200       | 0.8845               | 0.0193                | Mann-Whitney      | $2.04 \times 10^{-11}$ |
| 400       | 0.0112               | 0.4451                | Mann-Whitney      | $5.47 \times 10^{-11}$ |
| 800       | 0.6487               | 0.1659                | t-teste           | $4.05 \times 10^{-17}$ |
| 1600      | 0.2945               | 0.5287                | t-teste           | $1.84 \times 10^{-14}$ |



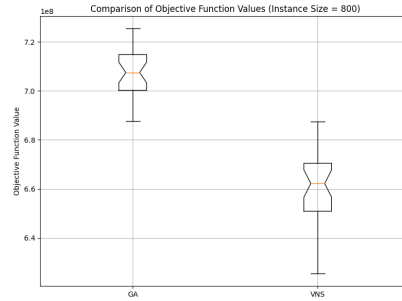
(a)  $I, J, K, E, S, Q, N1, N2, N3, M = 100$



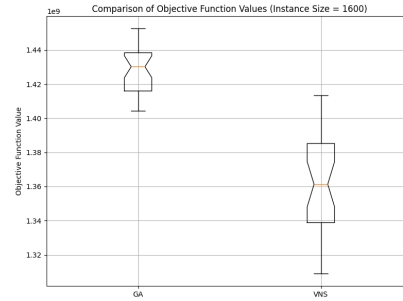
(b) = 200



(c) = 400



(d) = 800



(e) = 1600

Figura 5: Boxplots comparando o desempenho do algoritmo VNS e de uma meta-heurística baseada em população (GA) em diferentes tamanhos de problema, com base nos valores da função objetivo.

múltiplas etapas. Ao integrar produção, transporte e operações das instalações, o modelo garante uma alocação eficiente de recursos, respeitando as restrições de capacidade e demanda. Formulado como um problema de Programação Linear Inteira Mista (MILP), ele é resolvido por meio do algoritmo VNS, que explora o espaço de soluções utilizando estruturas de vizinhança como *Swap*, *Reversion*, *Insertion* e *Slide*.

A análise comparativa demonstra a superioridade do VNS

em relação ao GA, tanto em qualidade das soluções quanto em eficiência computacional, especialmente para instâncias de grande escala, onde métodos exatos se tornam inviáveis devido a limitações de tempo e memória.

Reconhece-se que a comparação entre os algoritmos pode ser influenciada pelos critérios de parada adotados. Embora ambos os métodos utilizem o mesmo número máximo de avaliações da função objetivo como critério principal, o VNS

possui um critério adicional de parada antecipada quando todos os operadores de vizinhança falham em melhorar a solução atual consecutivamente. Essa diferença metodológica pode contribuir para os menores tempos de execução observados no VNS, sugerindo maior eficiência na convergência. Como trabalho futuro, recomenda-se a investigação de critérios de parada unificados para permitir comparações ainda mais rigorosas entre as abordagens.

Como perspectiva de melhoria adicional, pesquisas futuras podem explorar operadores de vizinhança específicos para o problema, aprimorando ainda mais o desempenho do VNS. Além disso, uma abordagem híbrida, combinando VNS com métodos exatos, pode melhorar tanto a precisão quanto a eficiência da solução, tornando a metodologia mais aplicável a diferentes setores.

O código e conjunto de dados utilizados neste estudo estão disponíveis em: [https://github.com/raissagd/Pistachio\\_CLSC](https://github.com/raissagd/Pistachio_CLSC).

## REFERÊNCIAS

- [1] M. Gen, F. Altıparmak, and L. Lin, "A genetic algorithm for two-stage transportation problem using priority-based encoding," *OR Spectrum*, vol. 28, pp. 337–354, 2006.
- [2] A. Rajabi-Kafshgar, F. Gholian-Jouybari, I. Seyedi, and M. Hajiaghahi-Keshteli, "Utilizing hybrid metaheuristic approach to design an agricultural closed-loop supply chain network," *Expert Syst. Appl.*, vol. 217, p. 119504, 2023.
- [3] M. K. Doula, P. Kouloubis, and K. Elaiopoulos, "Turning wastes into valuable materials: Valorization of pistachio wastes in agricultural sector," in *SYMBIOSIS Int. Conf.*, pp. 19–21, 2014.
- [4] G. Bartzas and K. Komnitsas, "Life cycle analysis of pistachio production in Greece," *Sci. Total Environ.*, vol. 595, pp. 13–24, 2017.
- [5] R. Casper and E. Sundin, "Reverse logistic transportation and packaging concepts in automotive remanufacturing," *Procedia Manuf.*, vol. 25, pp. 154–160, 2018.
- [6] A. Cheraghalipour and E. Roghanian, "A bi-level model for a closed-loop agricultural supply chain considering biogas and compost," *Environ. Dev. Sustain.*, pp. 1–47, 2022.
- [7] K. Devika, A. Jafarian, and V. Nourbakhsh, "Designing a sustainable closed-loop supply chain network based on triple bottom line approach: A comparison of metaheuristics hybridization techniques," *Eur. J. Oper. Res.*, vol. 235, no. 3, pp. 594–615, 2014.
- [8] M. Zohal and H. Soleimani, "Developing an ant colony approach for green closed-loop supply chain network design: a case study in gold industry," *J. Clean. Prod.*, vol. 133, pp. 314–337, 2016.
- [9] E. K. Burke, J. Li, and R. Qu, "A hybrid model of integer programming and variable neighbourhood search for highly-constrained nurse rostering problems," *Eur. J. Oper. Res.*, vol. 203, no. 2, pp. 484–493, 2010.
- [10] C. C. Ribeiro, D. Aloise, T. F. Noronha, C. Rocha, and S. Urrutia, "An efficient implementation of a VNS/ILS heuristic for a real-life car sequencing problem," *Eur. J. Oper. Res.*, vol. 191, no. 3, pp. 596–611, 2008.
- [11] M. S. Pishvae, R. Z. Farahani, and W. Dullaert, "A memetic algorithm for bi-objective integrated forward/reverse logistics network design," *Comput. Oper. Res.*, vol. 37, no. 6, pp. 1100–1112, 2010.
- [12] F. Altıparmak, M. Gen, L. Lin, and T. Paksoy, "A genetic algorithm approach for multi-objective optimization of supply chain networks," *Comput. Ind. Eng.*, vol. 51, no. 1, pp. 196–215, 2006.
- [13] J. Liu, B. Sun, G. Li, and Y. Chen, "An integrated scheduling approach considering dispatching strategy and conflict-free route of AMRs in flexible job shop," *Int. J. Adv. Manuf. Technol.*, vol. 127, no. 3, pp. 1979–2002, 2023.
- [14] A. Imran, S. Salhi, and N. A. Wassan, "A variable neighborhood-based heuristic for the heterogeneous fleet vehicle routing problem," *Eur. J. Oper. Res.*, vol. 197, no. 2, pp. 509–518, 2009.
- [15] K. Fleszar and K. S. Hindi, "An effective VNS for the capacitated p-median problem," *Eur. J. Oper. Res.*, vol. 191, no. 3, pp. 612–622, 2008.
- [16] M. Rabiee, M. Zandieh, and A. Jafarian, "Scheduling of a no-wait two-machine flow shop with sequence-dependent setup times and probable rework using robust meta-heuristics," *Int. J. Prod. Res.*, vol. 50, no. 24, pp. 7428–7446, 2012.
- [17] B. Vahdani and M. Zandieh, "Scheduling trucks in cross-docking systems: Robust meta-heuristics," *Comput. Ind. Eng.*, vol. 58, no. 1, pp. 12–24, 2010.
- [18] M. S. Pishvae, J. Razmi, and S. A. Torabi, "An accelerated Benders decomposition algorithm for sustainable supply chain network design under uncertainty: A case study of medical needle and syringe supply chain," *Transp. Res. Part E: Logist. Transp. Rev.*, vol. 67, pp. 14–38, 2014.
- [19] S. Ahmadi and S. H. Amin, "An integrated chance-constrained stochastic model for a mobile phone closed-loop supply chain network with supplier selection," *J. Clean. Prod.*, vol. 226, pp. 988–1003, 2019.
- [20] M. Banasik, A. Stanisławska-Sachadyn, and P. Sachadyn, "A simple modification of PCR thermal profile applied to evade persisting contamination," *J. Appl. Genet.*, vol. 57, pp. 409–415, 2016.
- [21] F. Gholian-Jouybari, O. Hashemi-Amiri, B. Mosallanezhad, and M. Hajiaghahi-Keshteli, "Metaheuristic algorithms for a sustainable agri-food supply chain considering marketing practices under uncertainty," *Expert Syst. Appl.*, vol. 213, p. 118880, 2023.
- [22] Statista, "Leading producers of pistachios worldwide in 2022 and 2023," 2023. [Online]. Available: <https://www.statista.com/statistics/933042/global-pistachio-production-by-country/>
- [23] Y. Xie, Y. Sheng, M. Qiu, and F. Gui, "An adaptive decoding biased random key genetic algorithm for cloud workflow scheduling," *Eng. Appl. Artif. Intell.*, vol. 112, p. 104879, 2022.
- [24] Fathollahi-Fard, A. M., & Hajiaghahi-Keshteli, M. (2018). A stochastic multi-objective model for a closed-loop supply chain with environmental considerations. *Applied Soft Computing*, 69, 232–249.
- [25] Hokmabadi, H. (2018). Pistachio wastes in Iran and the potential to recapture them in value chain. *Pistachio and Health Journal*, 1(4), 1–12.
- [26] Dronavalli, S., & Kang, M. S. (2019). Microgametophytic selection for identifying maize with resistance to *Aspergillus spp.* and aflatoxin. *Journal of Crop Improvement*, 33(3), 363–371.
- [27] M. Gendreau and J.-Y. Potvin, Eds., *Handbook of Metaheuristics*, 2nd ed. Springer, 2010.
- [28] P. Hansen, N. Mladenović, and J. A. Moreno Perez, "Variable neighbourhood search: methods and applications," *Ann. Oper. Res.*, vol. 175, pp. 367–407, 2010.
- [29] Campelo, F., & Aranha, C. (2023). Lessons from the Evolutionary Computation Bestiary. *Artificial Life*, 29(4), 421–432. [https://doi.org/10.1162/artl\\_a\\_00402](https://doi.org/10.1162/artl_a_00402)
- [30] Aranha, C., Camacho Villalón, C. L., Campelo, F., Dorigo, M., Ruiz, R., Sevaux, M., Sörensen, K., & Stützle, T. (2022). Metaphor-based metaheuristics, a call for action: the elephant in the room. *Swarm Intelligence*, 16(1), 1–6.
- [31] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2024. [Online]. Available: <https://www.gurobi.com>