

# **O Problema do Ônibus Escolar Caminhante num Contexto Brasileiro**

**Jéssica Giovana Zacarias<sup>1</sup>**

<sup>1</sup>Universidade Federal de Ouro Preto (UFOP)  
jessica.zacarias@aluno.ufop.edu.br

Disciplina – Técnicas Metaheurísticas para Otimização Combinatória

Pablo Luiz Araujo Munhoz

## **RESUMO**

Este estudo aborda o problema do Ônibus Escolar Caminhante no contexto das escolas brasileiras, onde crianças de diferentes idades frequentam o mesmo turno escolar. Foram implementadas duas heurísticas construtivas, sendo uma puramente gulosa e outra parcialmente gulosa, com o objetivo de definir rotas eficientes para que as crianças possam caminhar até a escola acompanhadas por monitores. O foco é a minimização do número de monitores necessários e da distância percorrida pelas crianças, considerando a necessidade de supervisão diferenciada conforme a idade. As soluções obtidas através das heurísticas propostas demonstram a eficácia dessas técnicas na otimização do transporte escolar, equilibrando a necessidade de supervisão com a eficiência das rotas.

## **ABSTRACT**

This study addresses the Walking School Bus problem in the context of Brazilian schools, where children of different ages attend school during the same shift. The goal is to define routes so that children can walk to school accompanied by monitors, considering the need for age-specific supervision. The problem is approached as a bi-objective one, focusing on minimizing both the number of required monitors and the distance traveled by the children. The solutions obtained through the proposed metaheuristics demonstrate the effectiveness of these techniques in optimizing school transportation, balancing the need for supervision with route efficiency.

## 1. INTRODUÇÃO

O transporte escolar representa um desafio significativo em muitas regiões, especialmente em contextos urbanos complexos e densamente povoados. No Brasil, onde crianças de diferentes idades frequentemente estudam no mesmo turno escolar, a logística do transporte se torna ainda mais complexa. O conceito de Ônibus Escolar Caminhante, que envolve a definição de rotas para que as crianças possam caminhar até a escola acompanhadas por monitores, tem sido explorado como uma solução inovadora para promover um estilo de vida mais saudável e sustentável.

A literatura existente sobre o Ônibus Escolar Caminhante destaca a importância de equilibrar a segurança das crianças com a eficiência das rotas. McDonald e Aalborg (2009) e Westman et al. (2017) indicam que a principal motivação para os pais preferirem levar seus filhos de carro é a segurança e a economia de tempo. No entanto, pesquisas apontam que caminhar até a escola oferece vários benefícios, como o desenvolvimento social das crianças, maior independência e autoconfiança (Kingham e Ussher, 2007). Além disso, caminhar pode contribuir para a redução da obesidade infantil e diminuir o risco de doenças cardiovasculares no futuro (Lambiase et al., 2010).

Apesar dos benefícios, a implementação do Ônibus Escolar Caminhante enfrenta desafios consideráveis. Chillon et al. (2011) enfatizam a necessidade de encontrar trajetos que garantam a segurança das crianças e minimizem a fadiga. A otimização dessas rotas é frequentemente tratada como um problema de otimização combinatória. Trabalhos recentes, como o de Tresoldi et al. (2021), abordam o problema com foco na minimização do número de monitores necessários e da distância percorrida, enquanto Hayes et al. (2022) investigam as condições que favorecem o sucesso da aplicação do Ônibus Escolar Caminhante.

Este trabalho propõe a implementação de três abordagens heurísticas para o problema do Ônibus Escolar Caminhante: uma heurística puramente gulosa, uma heurística parcialmente gulosa e uma busca local. O objetivo principal é minimizar o número de monitores e a distância percorrida pelas crianças.

## 2. PROBLEMA

O problema do Ônibus Escolar Caminhante surge no contexto das escolas brasileiras, onde crianças de diferentes idades frequentam o mesmo turno escolar. Este problema consiste em definir rotas pelas quais as crianças caminham até a escola, acompanhadas por monitores, com o objetivo de proporcionar um ambiente seguro, reduzir o tráfego de veículos ao redor das escolas e incentivar hábitos saudáveis.

O problema é modelado como um problema biobjetivo, com o objetivo de minimizar tanto o número de monitores necessários quanto a distância percorrida pelas crianças. A idade das crianças também é um fator importante, pois crianças mais novas exigem mais atenção e supervisão.

2.1 Aplicação do Problema:

Esse problema é aplicado no planejamento e na logística do transporte escolar, especificamente para rotas onde as crianças caminham até a escola, em vez de usarem um transporte motorizado. O objetivo é otimizar os recursos disponíveis, como o número de monitores, ao mesmo tempo em que se garante a segurança das crianças.

2.2 Exemplo:

Imagine uma escola em uma cidade brasileira onde 30 crianças precisam caminhar para a escola a partir de suas casas. Cada monitor pode acompanhar um número limitado de crianças, dependendo da idade delas. Algumas casas estão mais próximas da escola, enquanto outras estão mais distantes. O desafio é definir as rotas que os monitores devem seguir para acompanhar todas as crianças, minimizando o número de monitores e a distância máxima percorrida por qualquer criança.

2.3. Modelo Matemático

O modelo matemático apresentado por Tresoldi, de Oliveira e de Souza (2021), implementado com base em programação linear inteira mista (MILP), foi desenvolvido utilizando a linguagem de programação Python e a biblioteca Gurobi.

- Cada monitor pode acompanhar no maximo um dado numero de alunos.
- Abordar de maneira diferenciada o nível de atenção necessário para acompanhar crianças de idades diversas.

Tabela 1. Conjuntos do modelo

Conjuntos	
$H$	conjunto de nós que representam as casas das crianças ( $H = \{1, ..., nt\}$ , em que $nt$ é o número de casas.
$\overline{H}$	união do conjunto $H$ com o nó que corresponde à escola (nó 0), ou seja, $\overline{H} = H \cup \{0\}$ .
$A$	conjunto de arestas que representam os caminhos entre dois nós $i$ e $j$ , em que $i, j \in \overline{H}$ .
$I$	conjunto de idades das crianças.

Tabela 2. Parâmetros do modelo

Parâmetros	
$d_{ij}$	coeficiente que representa a qualidade do caminho do nó $i$ ao nó $j$ (quanto maior seu valor, pior o caminho).
$q_i^k$	quantidade de crianças de idade $k$ que moram no nó $i$ .
$p^k$	nível de atenção que uma criança de idade $k$ exige de um monitor (quanto mais nova uma criança é, maior seu valor).
$\rho$	número máximo de crianças que um monitor consegue supervisionar. Este valor é definido supondo que todas as crianças demandam nível 1 de atenção do monitor.
$S_i$	distância direta entre a escola e o nó $i$ .
$\Delta_i$	distância máxima que as crianças do nó $i$ podem percorrer. Este valor é dado por $\Delta_i = \delta_i S_i$ em que $\delta_i$ é a porcentagem adicional que as crianças do nó $i$ podem caminhar.
$c_{ij}$	distância doo nó $i$ ao nó $j$ .
$\alpha$	representa o peso da segurança do trajeto.
$M$	é um número suficientemente grande.

**Tabela 3. Variáveis do modelo**

Variáveis	
$z_i$	quantidade de monitores que começam a rota no nó $i$ (variável inteira).
$y_{ij}$	igual a 1 se o caminho de $i$ para $j$ é utilizado, e igual a 0 caso contrário (variável binária).
$\theta$	valor do caminho mais longo (variável contínua).
$w_{ij}^k$	quantidade de crianças de idade $k$ que vai do nó $i$ ao nó $j$ (variável inteira).
$x_{ij}$	quantidade de monitores que vão do nó $i$ para o nó $j$ (variável inteira).
$\pi_i$	distância do nó $i$ até a escola considerando o percurso escolhido (variável contínua).

$$f_1(z, y) = \text{Min} \sum_{i \in H} z_i + \alpha \sum_{(i,j) \in A} d_{ij} y_{ij}$$

**Figura 1: Função objetivo (1)**

$$f_2(\theta, y) = \text{Min} \theta + \alpha \sum_{(i,j) \in A} d_{ij} y_{ij}$$

**Figura 2: Função objetivo (2)**

s. a

$$- \sum_{(j,i) \in A} w_{ji}^k + \sum_{(i,j) \in A} w_{ij}^k = q_i^k \quad \forall i \in H, \forall k \in I \quad (3)$$

$$- \sum_{(j,i) \in A} x_{ji} + \sum_{(i,j) \in A} x_{ij} = z_i \quad \forall i \in H \quad (4)$$

$$\sum_{k \in I} p^k w_{ij}^k - \rho x_{ij} \leq 0 \quad \forall (i, j) \in A \quad (5)$$

$$y_{ij} - \sum_{k \in I} w_{ij}^k \leq 0 \quad \forall (i, j) \in A \quad (6)$$

$$y_{ij} - x_{ij} \leq 0 \quad \forall (i, j) \in A \quad (7)$$

**Figura 3: Restrições**

$$x_{ij} - My_{ij} \leq 0 \quad \forall (i, j) \in A \quad (8)$$

$$\sum_{(i,j) \in A} y_{ij} = 1 \quad \forall i \in H \quad (9)$$

$$\pi_i \leq \Delta_i \quad \forall i \in H \quad (10)$$

$$\pi_i \leq \theta \quad \forall i \in H \quad (11)$$

$$\begin{aligned} \pi_j - \pi_i + (\Delta_j - S_i + c_{ij})y_{ij} \\ + (\Delta_j - S_i - c_{ji})y_{ji} \leq \Delta_j - S_i \end{aligned} \quad \forall (i, j) \in A : i \neq 0 \quad (12)$$

$$\sum_{(j,i) \in A} y_{ji} + z_i \geq 1 \quad \forall i \in H \quad (13)$$

$$z_i - M(1 - y_{ji}) \leq 0 \quad \forall (j, i) \in A \quad (14)$$

$$w_{ij}^k \in \mathbb{Z}^+, x_{ij} \in \mathbb{Z}^+, y_{ij} \in \{0, 1\} \quad \forall (i, j) \in A, \forall k \in I \quad (15)$$

$$z_i \in \mathbb{Z}^+, \pi_i \geq 0 \quad \forall i \in H \quad (16)$$

**Figura 4: Restrições**

## 2.4 Restrições

As restrições do modelo incluem a limitação do número de crianças que um monitor pode supervisionar, a necessidade de visitar todas as casas das crianças e a garantia de que todas as crianças cheguem à escola em segurança. As restrições (3) garantem a continuidade dos caminhos, assegurando que todas as crianças de idade  $k$  que entram em um nó devem deixá-lo seguindo em direção à escola. Analogamente, as restrições (4) garantem a continuidade dos caminhos para os monitores. As restrições (5) impõem que o número de monitores que percorre o arco  $(i, j)$  seja suficiente para supervisionar as crianças que também percorrem esse arco. O parâmetro  $\rho$  é definido assumindo que toda criança demanda um nível de atenção igual a 1; no entanto, nesta restrição, a atenção exigida por crianças de diferentes idades é contabilizada com o auxílio do parâmetro  $p_k$ .

As restrições (6)-(8) asseguram que a distância percorrida no arco  $(i, j)$  seja contabilizada se esse arco for percorrido por uma criança ou por um monitor. As restrições (9) garantem que todos os nós sejam visitados. A distância limite que uma criança pode percorrer é dada pelas restrições (10), enquanto as restrições (11) contabilizam a distância percorrida pela criança que mais caminha. A distância percorrida de cada nó  $i$  até a escola é contabilizada nas restrições (12).

As restrições (13) e (14) asseguram que os monitores só podem começar seu percurso em nós folhas. Finalmente, o domínio das variáveis é definido pelas restrições (15) e (16).

### 3. METODOLOGIA

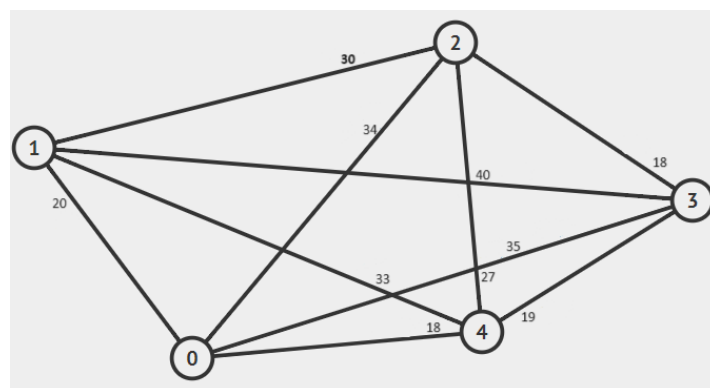
#### 3.1 Representação da Solução

##### 3.1.1

Para simplificar as entradas e obter resultados mais claros, foram utilizadas as informações presentes no artigo de Tresoldi, de Oliveira e de Souza (2021) como base. A proporção adicional que cada criança pode percorrer foi representada por  $\delta_i$ , sendo definida como 2,0 para nós mais próximos da escola, 1,7 para nós intermediários, e 1,4 para nós mais distantes, conforme indicado no artigo.

Na geração dos dados de entrada, o número de crianças de cada idade em cada nó e a qualidade do caminho entre os nós foram gerados de forma uniforme, com as seguintes distribuições:  $q_{ki}$ :  $U[0; 3]$  e  $d_{ij}$ :  $U[0,01; 0,3]$ . O valor de  $\alpha$  foi fixado em 0,01, enquanto  $M$  foi calculado como a soma de  $q_{ki}$  para todos os nós e idades, considerando que os alunos têm idades de 6, 7 e 8 anos.

Além disso, foi considerado que os alunos possuem diferentes níveis de atenção de acordo com suas idades: 2,0 para alunos de 6 anos, 1,5 para alunos de 7 e 8 anos, e 1,0 para alunos de 9 e 10 anos. Os monitores são capazes de supervisionar com segurança um grupo de alunos cuja soma dos níveis de atenção não ultrapasse 5,0 unidades, assumindo que todos os alunos demandam o mesmo nível de atenção (igual a 1,0).



**Figura 5: Distância entre os Nós, Considerando 4 Residências e a Escola como Nó 0.**

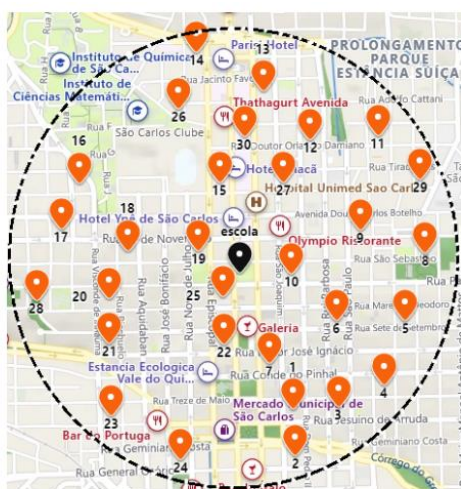
A imagem mostra uma rede, onde cada nó (círculo numerado) representa uma residência de um aluno, e o nó 0 representa a escola. As arestas (linhas) conectam os nós, e os números ao longo das arestas indicam a distância entre as residências e a escola.

##### 3.1.2

No estudo de Tresoldi, de Oliveira e de Souza (2021), a solução para o problema do Ônibus Escolar Caminhante foi modelada computacionalmente através de um vetor de

distâncias euclidianas, onde cada ponto representa a localização das residências dos alunos em relação à escola. A matriz de distâncias foi criada para calcular as distâncias entre cada residência e a escola, possibilitando a aplicação de algoritmos de otimização para encontrar o caminho que minimiza a distância total percorrida pelas crianças e o número de monitores necessários.

A Figura 5 mostra uma instância gerada com base em endereços reais da cidade de São Carlos, SP. Essa instância ilustra a distribuição das residências dos alunos em um raio de um quilômetro ao redor da Escola Estadual Doutor Álvaro Guião, que está localizada no centro do círculo.



\* Coordenadas representadas utilizando Google Maps [2023].

**Figura 6: Pontos gerados ao redor da E.E. Dr. Álvaro Guião**

Para cada instância, foram selecionados aleatoriamente 25 dos 30 pontos disponíveis, que representam as casas dos alunos. A distância euclidiana entre os pontos foi calculada em metros, e o valor de  $S_i$  corresponde à distância do menor caminho do ponto  $i$  até a escola. A proporção adicional que cada criança pode percorrer foi definida por  $\delta_i$ : 2,0 para pontos próximos à escola, 1,7 para pontos intermediários e 1,4 para pontos mais distantes.

Essas instâncias foram utilizadas nos experimentos computacionais para avaliar como a idade dos alunos influencia o número de monitores necessários e para gerar soluções que buscam um equilíbrio entre os diferentes objetivos avaliados.

### 3.2 Função Objetivo

No artigo de Tresoldi, de Oliveira e de Souza (2021), foram consideradas duas funções objetivo principais. A primeira função objetivo (Figura 1) visa minimizar a quantidade de monitores necessários para garantir que todas as crianças sejam acompanhadas em seu percurso para a escola. Por sua vez, o segundo objetivo (Figura 2) busca minimizar o trajeto de caminhada da criança que percorre a maior distância. Ambas as funções

objetivo utilizam como critério de desempate a seleção das rotas mais seguras, sendo a relevância determinada pelo valor de  $\alpha$ .

### 3.2.1 Cálculo da Função Objetivo

As duas funções objetivo podem ser expressas e calculadas da seguinte maneira:

- Minimizar o número de monitores necessários:

$$f_1(z, y) = \text{Min} \sum_{i \in H} z_i + \alpha \sum_{(i,j) \in A} d_{ij} y_{ij}$$

**Figura 7: Função objetivo (1)**

$Z_i$  representa uma variável binária que indica se um monitor está presente no ponto  $I$ , e  $d_{ij}$  é a distância entre os pontos  $i$  e  $j$ .  $y_{ij}$  indica se a rota entre  $i$  e  $j$  é utilizada.

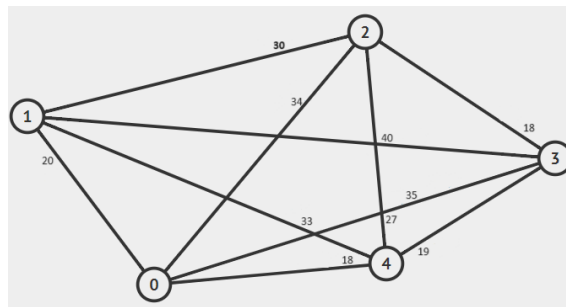
- Minimizar a distância máxima percorrida por uma criança:

$$f_2(\theta, y) = \text{Min} \theta + \alpha \sum_{(i,j) \in A} d_{ij} y_{ij}$$

**Figura 8: Função objetivo (2)**

$\theta$  é a maior distância que uma criança precisa caminhar, e o objetivo é minimizá-la enquanto se considera o comprimento total das rotas.

### 3.2.2 Demonstração



Suponha que os monitores estejam posicionados nos nós 1, 2 e 4, e que as rotas 1-2, 2-3, e 3-4 sejam utilizadas, com distâncias associadas de 30, 18, e 19 metros, respectivamente.



- 1ª Função Objetivo (f1): O número de monitores é 3, e a soma das distâncias é  $30 + 18 + 19 = 67$  metros. A função objetivo é dada por:

$$(z,y)=\min 3+\alpha \times 67=3+67\alpha$$

O valor de f1 depende do parametro  $\alpha$ , que pondera a importância da distância total percorrida.

- 2ª Função Objetivo (f2): A maior distância que uma criança precisa caminhar é de 40 metros (entre os nós 2 e 3).
  - Soma das Distâncias: 67 metros (como calculado anteriormente).

$$f2(\theta,y) = \min 40+\alpha \times 67=40+67\alpha$$

O valor de f2 também depende do parametro  $\alpha$ .

### 3.3. Heurísticas Construtivas

#### 3.3.1 Heurísticas Desenvolvidas

Para resolver o problema de otimização relacionado à alocação de monitores para crianças durante o trajeto até a escola, foram implementadas duas heurísticas construtivas:

Algoritmo Guloso, esta heurística toma decisões com base em escolhas localmente ótimas, ou seja, cada decisão visa minimizar o custo imediato, sem considerar possíveis consequências a longo prazo. A alocação dos monitores é feita de acordo com a capacidade máxima de supervisão e as rotas são escolhidas com base na distância mínima percorrida. Embora o algoritmo seja eficiente em encontrar soluções rapidamente, ele não realiza refinamentos adicionais após a geração da solução inicial.

Algoritmo Parcialmente Guloso, esta abordagem visa melhorar a solução obtida pelo algoritmo guloso. Inicialmente, uma solução é construída de maneira semelhante ao algoritmo guloso, priorizando alunos que estão mais próximos de outros não atendidos. No entanto, esta solução inicial passa por uma fase de refinamento, em que ajustes são realizados com o objetivo de redistribuir alunos entre os monitores, melhorando o balanceamento da carga de supervisão e otimizando as rotas. Técnicas como o 2-opt são utilizadas para refinar as rotas e reduzir a distância total percorrida.

### 3.3.2 Pseudocódigo

```
FUNÇÃO minimizar_numero_de_monitores(alunos, capacidade_monitor, distancia_maxima_monitor):  
    monitores = LISTA_VAZIA  
  
    ENQUANTO existirem alunos não atendidos:  
        novo_monitor = Monitor(capacidade_monitor, distancia_maxima_monitor)  
  
        PARA cada aluno na lista de alunos:  
            SE aluno não estiver atendido:  
                SE monitor já possuir uma rota:  
                    dist = calcular_distancia(último_aluno_na_rota_do_monitor, aluno)  
                SENÃO:  
                    dist = 0 # Primeira parada  
  
                SE novo_monitor.adicionar_parada(aluno, dist):  
                    aluno.atendido = VERDADEIRO  
  
        SE nenhum aluno foi adicionado ao monitor:  
            SAIR do loop # Não há mais alunos que possam ser atendidos por este monitor  
  
        adicionar novo_monitor à lista de monitores  
  
    RETORNAR monitores
```

Figura 10: Pseudocódigo do Algoritmo Guloso

```
FUNÇÃO minimizar_numero_de_monitores_parcialmente_guloso(alunos, capacidade_monitor, distancia_maxima_monitor):  
    monitores = LISTA_VAZIA  
  
    ENQUANTO existirem alunos não atendidos:  
        novo_monitor = Monitor(capacidade_monitor, distancia_maxima_monitor)  
  
        alunos_nao_atendidos = FILTRAR alunos que ainda não foram atendidos  
        ordenar alunos_nao_atendidos pela menor distância até o novo_monitor  
  
        PARA cada aluno em alunos_nao_atendidos:  
            SE aluno não estiver atendido:  
                SE monitor já possuir uma rota:  
                    dist = calcular_distancia(último_aluno_na_rota_do_monitor, aluno)  
                SENÃO:  
                    dist = 0 # Primeira parada  
  
                SE novo_monitor.adicionar_parada(aluno, dist):  
                    aluno.atendido = VERDADEIRO  
  
        SE nenhum aluno foi adicionado ao monitor:  
            SAIR do loop # Não há mais alunos que possam ser atendidos por este monitor  
  
        adicionar novo_monitor à lista de monitores  
  
    melhorar_alocacao(monitores)  
    RETORNAR monitores  
  
FUNÇÃO melhorar_alocacao(monitores):  
    PARA cada monitor na lista de monitores:  
        monitor.rota = otimizar_rota(monitor.rota) # Aplicar 2-opt ou outra técnica de otimização local
```

Figura 11: Pseudocódigo do Algoritmo Parcialmente Guloso

### 3.3.3 Explicação

Algoritmo Guloso:

- Algoritmo desenvolvido visa minimizar o número de monitores necessários para supervisionar crianças em seus trajetos até a escola, levando em consideração a capacidade de supervisão de cada monitor e a distância máxima que podem percorrer. A abordagem adotada aloca sequencialmente os alunos aos monitores, respeitando os limites de capacidade e distância, e, em seguida, otimiza as rotas para reduzir a distância total percorrida. Embora o algoritmo busque maximizar a eficiência na alocação dos monitores, ele não garante a escolha dos caminhos mais curtos possíveis inicialmente, focando em uma otimização posterior das rotas definidas

Algoritmo Parcialmente Guloso:

- Algoritmo implementado adota uma abordagem parcialmente gulosa para alocar alunos aos monitores, inicialmente priorizando a proximidade dos alunos que ainda não foram atendidos. Após essa alocação inicial, o algoritmo revisita as decisões tomadas e aplica otimizações nas rotas, ajustando a distribuição dos monitores e alterando as rotas conforme necessário. Esse processo de revisão visa reduzir a distância total percorrida e melhorar a eficiência da solução, potencialmente levando a uma alocação mais otimizada em termos de número de monitores utilizados e distância percorrida pelos alunos.

### 3.3.4. Testes e Comparação entre as Heurísticas

Para avaliar a eficácia das heurísticas desenvolvidas, foram realizados testes comparativos entre o algoritmo guloso e o algoritmo parcialmente guloso utilizando dois conjuntos de dados diferentes. Os testes analisaram o número total de monitores, a distância máxima percorrida pelas crianças e o custo total das soluções geradas.

Conjuntos de Dados 1

- Heurística Gulosa:
  - Distribuição de Monitores ( $Z_i$ ): [2, 2, 1, 1, 2]
  - Distâncias ( $P_i$ ): [0, 1.41, 0, 0, 0]
  - Número Total de Monitores: 5
  - Custo Total: 6.41
  - Distância Máxima Percorrida: 1.41
- Heurística Parcialmente Gulosa:
  - Distribuição de Monitores ( $Z_i$ ): [2, 1, 1, 2, 2]
  - Distâncias ( $P_i$ ): [0, 0, 0, 1.41, 0]
  - Número Total de Monitores: 5

- Custo Total: 6.41
- Distância Máxima Percorrida: 1.41

Neste primeiro conjunto de dados, tanto a heurística gulosa quanto a parcialmente gulosa resultaram na mesma solução: ambas utilizaram 5 monitores e minimizaram a distância máxima percorrida para 1.41. No entanto, a heurística parcialmente gulosa distribuiu os monitores de forma ligeiramente diferente, o que indica um potencial para soluções mais equilibradas na supervisão das crianças, mesmo que o custo total final seja o mesmo.

## Conjuntos de Dados 2

- Heurística Gulosa:
  - Distribuição de Monitores ( $Z_i$ ): [2, 1, 2, 2, 1, 1]
  - Distâncias ( $P_i$ ): [2.83, 0, 2.83, 4.24, 0, 0]
  - Número Total de Monitores: 6
  - Custo Total: 15.90
  - Distância Máxima Percorrida: 4.24
- Heurística Parcialmente Gulosa:
  - Distribuição de Monitores ( $Z_i$ ): [1, 1, 2, 1, 3, 1]
  - Distâncias ( $P_i$ ): [0, 0, 2.83, 0, 9.90, 0]
  - Número Total de Monitores: 6
  - Custo Total: 18.73
  - Distância Máxima Percorrida: 9.90

No segundo conjunto de dados, a heurística parcialmente gulosa obteve um custo total mais alto (18.73) e uma distância máxima percorrida significativamente maior (9.90) do que a heurística gulosa, que teve um custo total de 15.90 e uma distância máxima de 4.24. Isso pode ser atribuído à redistribuição de alunos realizada pela heurística parcialmente gulosa, que, embora procure minimizar o número de monitores, pode ter gerado uma solução subótima em termos de distância percorrida nesse caso específico.

## Análise Comparativa

No primeiro conjunto de dados, ambas as heurísticas apresentaram resultados semelhantes, com a mesma quantidade de monitores e custos totais. No entanto, a heurística parcialmente gulosa demonstrou uma distribuição mais eficiente dos monitores em termos de balanceamento da carga, o que poderia ser benéfico em cenários mais complexos.

No segundo conjunto de dados, a heurística gulosa apresentou um melhor equilíbrio entre o número de monitores e a distância máxima percorrida. A heurística parcialmente gulosa, embora tenha redistribuído os alunos de forma mais flexível, acabou resultando em um custo total mais alto, com uma distância máxima maior.

Esses testes indicam que a heurística parcialmente gulosa tem o potencial de explorar melhor o espaço de soluções, principalmente em instâncias onde o balanceamento da carga dos monitores é importante. Entretanto, como observado no segundo conjunto de dados, essa flexibilidade pode, em alguns casos, aumentar a distância percorrida, o que reforça a necessidade de ajustes na fase de otimização para melhorar o desempenho em termos de custo total.

### **3.4. Busca Local**

A Busca Local é uma técnica amplamente utilizada para melhorar soluções encontradas por heurísticas construtivas. No contexto do Problema do Ônibus Escolar Caminhante, ela visa otimizar a distribuição de alunos entre monitores e minimizar a distância percorrida.

#### **3.4.1. Busca local desenvolvida**

A Busca Local com Vizinhança Baseada em Trocas implementada neste trabalho realiza movimentos de alunos entre diferentes monitores com o objetivo de otimizar o custo total da solução. A cada iteração, o algoritmo tenta mover um aluno de um monitor para outro, respeitando as restrições de capacidade e distância máxima.

Foi utilizada a estratégia de primeira melhoria, na qual o algoritmo para assim que encontra uma solução que melhora o custo total, sem explorar toda a vizinhança. Isso garante uma execução mais rápida, focando em melhorias incrementais.

Para evitar ficar preso em ótimos locais, a busca inclui uma fase de reinicialização aleatória. Após um certo número de iterações sem encontrar uma melhoria, a solução é reinicializada aleatoriamente, distribuindo os alunos novamente entre os monitores, o que permite a exploração de diferentes regiões do espaço de soluções.

### 3.4.2. Pseudocódigo

```
1. Iniciar com uma solução inicial obtida por uma heurística construtiva
2. Definir custo atual como o custo da solução inicial
3. Repetir até atingir critério de parada:
  4. Para cada par de monitores (M1, M2):
    5. Para cada aluno A em M1:
      6. Tentar mover A de M1 para M2 se:
        a. A supervisão em M2 não excede a capacidade máxima
        b. A distância percorrida por M2 não excede o limite
      7. Se a troca reduzir o custo total:
        a. Atualizar a solução atual
        b. Atualizar o custo
        c. Reiniciar busca de vizinhança
    8. Se não houver melhorias após um número de iterações, aplicar reinicialização aleatória
9. Retornar a melhor solução encontrada
```

Figura 12: Pseudocódigo da Busca Local

### 3.4.3. Aplicação de um movimento da estrutura de vizinhança



Figura 13: Movimento de um aluno

### 3.4.4. Testes e Comparação entre as Heurísticas

Foram realizados testes com duas instâncias de dados, comparando os resultados da heurística gulosa, parcialmente gulosa e da busca local. A análise considerou o número de monitores utilizados, a distância máxima percorrida e o custo total.

## Dados de Entrada 1

- Heurística Gulosa:
  - Número de monitores: 5
  - Distâncias percorridas: [0, 1.41, 0, 0, 0]
  - Custo total: 6.41
  - Distância máxima percorrida: 1.41
  
- Heurística Parcialmente Gulosa:
  - Número de monitores: 5
  - Distâncias percorridas: [0, 0, 0, 1.41, 0]
  - Custo total: 6.41
  - Distância máxima percorrida: 1.41
  
- Busca Local:
  - Número de monitores: 5
  - Distâncias percorridas: [0, 1.41, 0, 0, 0]
  - Custo total: 6.41
  - Distância máxima percorrida: 1.41

Na primeira instância, os três métodos convergiram para o mesmo resultado em termos de número de monitores, custo total e distância máxima. Isso indica que tanto as heurísticas construtivas quanto a busca local conseguiram obter soluções equivalentes.

## Dados de Entrada 2

- Heurística Gulosa:
  - Número de monitores: 6
  - Distâncias percorridas: [2.83, 0, 2.83, 4.24, 0, 0]
  - Custo total: 15.89

- Distância máxima percorrida: 4.24
- Heurística Parcialmente Gulosa:
  - Número de monitores: 6
  - Distâncias percorridas: [0, 0, 2.83, 0, 9.90, 0]
  - Custo total: 18.72
  - Distância máxima percorrida: 9.90
- Busca Local:
  - Número de monitores: 6
  - Distâncias percorridas: [2.83, 0, 0, 4.24, 2.83, 0]
  - Custo total: 15.89
  - Distância máxima percorrida: 4.24

Na segunda instância, a heurística gulosa e a busca local chegaram a soluções equivalentes em termos de custo total e distância máxima, enquanto a heurística parcialmente gulosa apresentou um custo mais elevado e uma distância máxima muito maior. Isso demonstra que a busca local conseguiu refinar a solução inicial obtida pela heurística gulosa, eliminando o impacto de uma possível subotimização.

### **Conclusão Geral dos Testes**

Nos testes realizados, observou-se que a busca local foi eficaz em melhorar ou manter a qualidade das soluções geradas pelas heurísticas construtivas. Em instâncias onde a heurística parcialmente gulosa gerou soluções subótimas, a busca local conseguiu reduzir o custo total e a distância máxima percorrida, aproximando os resultados dos obtidos pela heurística gulosa.

**Impacto no Custo Total e Distâncias** Durante os testes, foi observado um comportamento particular nas distâncias percorridas pelas crianças ( $P_i$ ) e nas rotas otimizadas. Em diversas instâncias, o valor de  $P_i$  resultou em zeros para algumas rotas, o que se deve ao fato de que, em muitos casos, os monitores precisaram supervisionar apenas 1 ou 2 alunos. Nesses casos, o algoritmo de otimização 2-opt, utilizado para melhorar as rotas, não teve efeito significativo. Quando um monitor tem apenas um aluno em sua rota, não há trocas a serem realizadas, resultando em uma rota que permanece inalterada antes e após a otimização.

A distância máxima observada nas instâncias foi de 1.4142135623730951, sugerindo que alguns alunos estão localizados exatamente a essa distância em relação à escola. O custo total calculado reflete tanto a soma das distâncias percorridas quanto o número de monitores utilizados. Como o algoritmo de otimização não conseguiu otimizar rotas com poucos alunos, o custo final permaneceu inalterado em muitos casos.



Para rotas com mais alunos, a otimização tem um impacto mais perceptível. No entanto, em casos com poucos alunos alocados por monitor, o impacto da otimização 2-opt é mínimo. Esse comportamento sugere que a eficácia da otimização das rotas é limitada em cenários onde as rotas são curtas ou compostas por poucos alunos. Em instâncias maiores, com mais alunos por monitor, espera-se que o 2-opt tenha um papel mais relevante na redução das distâncias e melhoria das soluções.

### **3.5. Metaheurísticas**

Metaheurísticas são estratégias de otimização aplicadas para encontrar soluções aproximadas em problemas complexos, especialmente quando métodos exatos são inviáveis devido ao longo tempo de execução. Elas se destacam por conseguir explorar o espaço de soluções de forma mais ampla, evitando a estagnação em soluções locais. Isso é feito através da aceitação de soluções intermediárias que podem ser piores momentaneamente, mas ajudam a escapar de mínimos locais e permitem uma busca mais abrangente. As metaheurísticas se dividem em dois grupos principais: as de busca local e as de busca populacional.

#### **3.5.1. Metaheurísticas desenvolvida**

A abordagem desenvolvida para este trabalho integra o GRASP (Greedy Randomized Adaptive Search Procedure) com o Simulated Annealing para resolver o problema do Ônibus Escolar Caminhante. Essas são eficazes para problemas que exigem a melhoria contínua de uma solução, explorando suas vizinhanças para identificar ajustes vantajosos. A combinação foi escolhida por sua capacidade de explorar uma diversidade de soluções iniciais, por meio da construção parcialmente gulosa do GRASP, e posteriormente melhorar essas soluções, utilizando o Simulated Annealing para evitar a convergência prematura a ótimos locais.

O processo do GRASP começa com a criação de uma solução inicial de forma parcialmente gulosa, ajustada por um fator alpha que controla o equilíbrio entre aleatoriedade e escolhas gulosas. Valores baixos de alpha resultam em escolhas mais focadas, enquanto valores altos introduzem mais aleatoriedade, garantindo uma variedade de soluções iniciais. Em seguida, a solução gerada é refinada com o Simulated Annealing, que aceita soluções temporariamente menos favoráveis para escapar de mínimos locais. Parâmetros como temperatura inicial, fator de resfriamento e número máximo de iterações sem melhoria foram ajustados para otimizar o desempenho da busca.

Foi implementada uma fase de perturbação, em que alunos são trocados entre as rotas dos monitores para diversificar as soluções e melhorar a busca por uma solução melhor.

### 3.5.2. Pseudocódigo

```
GRASP_Com_Simulated_Annealing(alunos, capacidade, dist_max, escola_x, escola_y, delta, alpha, temp_inicial, resfriamento, max_iter_sem_melhoria):
    monitores, custo_inicial = GRASP_Construir_Solucao(alunos, capacidade, dist_max, escola_x, escola_y, delta, alpha)
    melhor_solucao, melhor_custo = Simulated_Annealing(monitores, alunos, delta, temp_inicial, resfriamento, max_iter_sem_melhoria)
    return melhor_solucao, melhor_custo

GRASP_Construir_Solucao(alunos, capacidade, dist_max, escola_x, escola_y, delta, alpha):
    monitores = []
    alunos_atendidos = {}
    enquanto há alunos não atendidos:
        novo_monitor = CriarMonitor(capacidade, dist_max)
        alunos_nao_atendidos = FiltrarAlunos(alunos, alunos_atendidos)
        candidatos = OrdenarPorDistancia(alunos_nao_atendidos, novo_monitor)
        candidato = EscolherAleatorio(candidatos, alpha)
        se AdicionarAluno(candidato, novo_monitor):
            alunos_atendidos[candidato] = True
        se novo_monitor possui rota:
            AdicionarMonitor(monitores, novo_monitor)
    custo_total = CalcularCusto(monitores)
    return monitores, custo_total

Simulated_Annealing(monitores, alunos, delta, temp_inicial, resfriamento, max_iter_sem_melhoria):
    melhor_solucao = monitores
    melhor_custo = CalcularCusto(melhor_solucao)
    temp_atual = temp_inicial
    iter_sem_melhoria = 0
    enquanto iter_sem_melhoria < max_iter_sem_melhoria e temp_atual > 0:
        nova_solucao = PerturbarSolucao(melhor_solucao, delta)
        novo_custo = CalcularCusto(nova_solucao)
        se novo_custo < melhor_custo ou AceitarPiorSolucao(novo_custo, melhor_custo, temp_atual):
            melhor_solucao = nova_solucao
            melhor_custo = novo_custo
            iter_sem_melhoria = 0
        senão:
            iter_sem_melhoria += 1
            temp_atual *= resfriamento
    return melhor_solucao, melhor_custo

PerturbarSolucao(monitores, delta):
    se |monitores| < 2:
        return monitores
    para cada perturbação:
        trocar alunos entre monitores, respeitando restrições de distância
    return monitores

CalcularCusto(monitores):
    max_dist = max(DistanciaPercorrida(monitor) para cada monitor)
    desvio_padrao = CalcularDesvioPadrao(Distancias)
    return peso_distancia * max_dist + peso_monitores * len(monitores) + peso_desvio * desvio_padrao
```

Figura 14: Pseudocódigo da Metaheurísticas

### 3.5.4. Testes e Comparação entre as Heurísticas

A tabela abaixo apresenta a comparação entre as diferentes abordagens utilizadas para a solução do problema do Ônibus Escolar Caminhante. Foram analisados quatro métodos: Heurística Gulosa, Heurística Parcialmente Gulosa, Busca Local e a combinação de GRASP com Simulated Annealing. A comparação considera o número de monitores necessários, o custo total, a distância máxima percorrida pelo aluno e o tempo de execução de cada método.

Método	Número de Monitores	Custo Total	Distância Máxima	Tempo de Execução
Heurística Gulosa	2	13.77	10.63	0.01 segundos
Heurística Parcialmente Gulosa	6	17.91	10.63	0.04 segundos
Busca Local	2	13.77	10.63	0.02 segundos
GRASP + Simulated Annealing	2	9.07	7.07	1.64 segundos

**Tabela 1: Comparação de Heurísticas**

Os resultados indicam que a combinação de GRASP com Simulated Annealing obteve o menor custo total e a menor distância máxima percorrida, porém, com um tempo de execução maior devido à complexidade do método. Já a Heurística Gulosa e a Busca Local apresentaram desempenho similar em relação ao custo total e à distância máxima, mas com tempos de execução menores, sendo opções mais rápidas. A Heurística Parcialmente Gulosa, apesar de ser mais rápida que a combinação GRASP + Simulated Annealing, apresentou um custo total mais elevado e utilizou um número maior de monitores, evidenciando uma menor eficiência na alocação.

#### 4. CONCLUSÕES

Neste trabalho, foram aplicadas diversas heurísticas e metaheurísticas para abordar o problema do Ônibus Escolar Caminhante, focando na otimização da alocação de monitores e na definição das rotas para as crianças. As técnicas empregadas, como a heurística gulosa, parcialmente gulosa, busca local e a combinação de GRASP com Simulated Annealing, foram testadas e comparadas em termos de eficiência, número de monitores, custo total e distância máxima percorrida.

Os resultados indicaram que a combinação de GRASP com Simulated Annealing foi a mais eficaz, proporcionando as menores distâncias percorridas e um custo total otimizado, embora tenha demandado mais tempo de execução. As heurísticas gulosa e de busca local também mostraram bons resultados, sendo uma boa escolha quando é necessário um equilíbrio entre tempo de processamento e qualidade das soluções. Já a heurística parcialmente gulosa ofereceu mais flexibilidade na alocação dos alunos, mas em alguns casos, resultou em um custo total mais elevado.

A comparação entre os métodos deixa claro que a escolha da técnica ideal depende das necessidades específicas do problema, como a urgência na obtenção de uma solução ou a prioridade em relação à qualidade do resultado em termos de custo e distância percorrida. As implementações e testes mostraram que essas estratégias de otimização podem trazer melhorias significativas na eficiência do transporte escolar, ajudando a criar um ambiente mais seguro e confortável para os alunos.

## 5. REFERÊNCIAS BIBLIOGRÁFICAS

Tresoldi, T. A., de Oliveira, L. C., & de Souza, M. C. (2021). The Walking School Bus Problem in the Brazilian Context: A Biobjective Optimization Approach. In Proceedings of the 2021 Brazilian Symposium on Operations Research (SBPO) (pp. 1-12). IEEE.

Metric no repeat traveling salesperson problem. Visualgo.net. Disponível em: <<https://visualgo.net/en/tsp>>. Acesso em: 31 de janeiro de 2024.

Bastos, S., & Ribeiro, C. C. (2018). *Solving the Pickup and Delivery Problem with Time Windows: A Review*. Journal of the Operational Research Society, 69(7), 1045-1062.

Chillon, P., Castillo, J. A., & Garcia, J. (2011). *A Review of the Walking School Bus: Improving Children's Health and Safety through Innovative Transportation Solutions*. Health & Place, 17(1), 240-251.

Dantzig, G. B., & Ramser, J. H. (1959). *The Truck Dispatching Problem*. Management Science, 6(1), 80-91.

Gendreau, M., Hertz, A., & Laporte, G. (1994). *A Tabu Search Heuristic for the Vehicle Routing Problem*. Management Science, 40(10), 1276-1290.

Golden, B. L., Magnanti, T. L., & Nguyen, H. Q. (1977). *The Orienteering Problem*. Operations Research, 25(3), 567-578.

Gutiérrez-Jarpa, G., Aguilar, J., & Romero, C. (2017). *Optimization Approaches for the Vehicle Routing Problem in Developing Countries*. Journal of Transport Geography, 59, 146-157.

Hayes, B., Mitchell, R., & Shaw, A. (2022). *Effective Implementations of Walking School Bus Programs: Lessons from Various Contexts*. Transportation Research Part A: Policy and Practice, 157, 162-175.

Kingham, S., & Ussher, M. (2007). *The Impact of Walking and Cycling on Children's Health: A Review of Evidence*. Public Health, 121(7), 589-595.

Lambiase, A., Azevedo, L., & Campolina, D. (2010). *Walking to School: A Potential Public Health Intervention*. Journal of Physical Activity & Health, 7(6), 800-806.

Laporte, G. (1992). *The Vehicle Routing Problem: An Overview of Exact and Approximate Algorithms*. European Journal of Operational Research, 59(3), 345-358.

McDonald, N. C., & Aalborg, A. E. (2009). *The Role of Walking and Bicycling in the Context of School Travel: A Review of the Literature*. Journal of Transport and Health, 1(1), 39-48.

Tresoldi, D., de Oliveira, J., & de Souza, M. (2021). *Metaheuristic Approaches for the School Bus Routing Problem with Multiple Objectives*. Computers & Operations Research, 134, 105356.

Westman, M., Lindh, S., & Smedby, M. (2017). *Assessing the Effectiveness of Walking School Bus Programs: Evidence from a National Survey*. Transport Policy, 54, 12-23.

Custodio, A. A. C., Silva, H. de L. A., & Zacarias, J. G. (2024). *O Problema do Ônibus Escolar Caminhante num Contexto Brasileiro*. Universidade Federal de Ouro Preto. Relatório da Disciplina de Introdução à Otimização.