

# Uma Implementação de Algoritmo Genético para o Problema do Roteamento de Veículos Capacitado (PRVC)

**Autor: Victor Lelis Soares**  
**Orientadora: Bianca de Almeida Dantas**

<sup>1</sup>Faculdade de Computação – Universidade Federal de Mato Grosso do Sul (UFMS)  
Caixa Postal 15.064 – 91.501-970 – Campo Grande – MS – Brasil

{victor.lelis@ufms.br}

{bianca.dantas@ufms.br}

## 1. Introdução

Grande parte do valor final dos produtos destinados aos clientes está diretamente ligada ao processo de distribuição, por isso, serviços baseados em entregas visam abranger o maior número possível de clientes para minimizar os custos operacionais [Silva Melo and Ferreira Filho 2001].

Nesse contexto, surge o Problema do Roteamento de Veículos (PRV), proposto por Dantzig e Ramser [Dantzig and Ramser 1959]. Os desafios reais são delineados por diversas restrições, como a capacidade dos veículos, janelas de tempo para atendimento de clientes e proibições em determinados caminhos. Essas restrições geram distintas variações do PRV, cada uma refletindo um cenário específico do mundo real. A presença dessas restrições solidifica o PRV como um dos problemas mais desafiadores no âmbito da otimização combinatória.

Vale destacar que o PRV é classificado como NP-difícil [Toth and Vigo 2002]. Um problema NP-difícil indica que não há algoritmo eficiente conhecido para resolver todas as instâncias do problema em tempo polinomial [Toth and Vigo 2002]. O PRV representa um desafio computacional substancial, motivando o desenvolvimento de uma variedade de heurísticas, meta-heurísticas e modelos híbridos para abordar eficazmente essas complexidades, uma vez que encontrar soluções ótimas pode ser impraticável em um tempo razoável.

O objetivo deste trabalho é apresentar o PRV e algumas de suas variações e propor uma solução para o Problema do Roteamento de Veículos Capacitado (PRVC) usando Algoritmos Genéticos (AGs).

O restante deste trabalho está organizado da seguinte maneira: a Seção 2 destaca alguns trabalhos relacionados, evidenciando a evolução do estado da arte, a Seção 3 aborda o referencial teórico para fundamentar o projeto, a Seção 4 expõe a metodologia adotada na implementação do trabalho, a Seção 5 engloba os experimentos realizados e a análise dos resultados obtidos e, por fim, a Seção 6 aborda a conclusão e os trabalhos futuros.

## 2. Trabalhos Relacionados

Nos últimos anos, tem ocorrido um aumento na proposição de meta-heurísticas para resolver o PRV [Laporte and Semet 2002, Braekers et al. 2016]. Estas meta-heurísticas são

abordagens genéricas que buscam soluções de alta qualidade, muitas vezes incorporando técnicas convencionais de construção e otimização de rotas. Uma característica marcante dessas meta-heurísticas é a sua capacidade de considerar soluções intermediárias que podem ser sub-ótimas ou até inviáveis durante o processo de busca, o que representa uma divergência significativa das abordagens clássicas [Laporte and Semet 2002].

Entre diversos tipos de meta-heurísticas aplicados ao PRV podem ser citadas: *Simulated Annealing* [Robust et al. 1990, Alfa et al. 1991, Osman 1993], *Deterministic Annealing* [Dueck and Scheuer 1990], Busca Tabu [Willard 1989, Pureza and França 1991, Osman 1993, Gendreau et al. 1994], Algoritmos Genéticos [Mulloorakam and Nidhiry 2019, Baker and Ayechew 2003, Wang and Lu 2009], *Ant Systems* [Colorni et al. 1991, Kawamura et al. 1998] e Redes Neurais [Durbin and Willshaw 1987, El Ghaziri 1993].

Os algoritmos genéticos podem resolver o PRVC e podem competir em qualidade de soluções e tempo com outras soluções baseadas em heurísticas [Baker and Ayechew 2003], tal qual Busca Tabu [Glover 1986] e *Simulated Annealing* [Kirkpatrick et al. 1983]. Ao longo dos anos, cada vez mais AGs híbridos tornaram-se populares [Braekers et al. 2016], devido à natureza complementar entre AGs e outras heurísticas [Wang and Lu 2009].

Em [Baker and Ayechew 2003], os autores utilizam um Algoritmo Genético (AG) híbrido para solucionar instâncias do PRVC, o AG é combinado com Algoritmo de Varredura [Gillett and Miller 1974] e uma estrutura de geração de soluções inspirada na abordagem de Fisher e Jaikumar [Fisher and Jaikumar 1981].

O primeiro método, o algoritmo de varredura, é um método heurístico que se concentra na construção de soluções iniciais viáveis para o PRV. A ideia chave por trás do algoritmo é organizar os clientes de acordo com um ângulo polar em relação ao depósito central e, em seguida, varrer esses clientes em ordem crescente de ângulo polar para alocá-los aos veículos [Gillett and Miller 1974].

No segundo método, os autores estão gerando uma população inicial de soluções em que a estrutura da rota pode ser bastante simplificada e em que um grau considerável de violação de restrições pode ocorrer. Consequentemente, simplificamos um pouco o método de geração dos pontos de partida e, em vez de resolver um problema de atribuição generalizada, usamos uma alocação aleatória para cada cliente a um de seus dois melhores locais de partida [Baker and Ayechew 2003].

Por outro lado, há pesquisas que visam medir o impacto dos parâmetros do AG na qualidade das soluções. Em [de Araujo Lima and de Araújo 2020], os autores realizam um estudo comparativo entre dois esquemas de representação, uma representação binária usando matriz e uma representação inteira usando vetor. A representação binária, por meio de matriz tridimensional, está organizada em: eixo das colunas ( $M$ ) que indica o cliente a ser visitado, eixo das linhas ( $N$ ) que indica a ordem de visita e eixo da profundidade ( $Z$ ) que indica o veículo que será designado para atender o cliente.

A representação inteira consiste em um vetor de inteiros de tamanho  $N$  representando a quantidade de clientes, e o valor associado a cada posição do vetor representa a demanda associada a cada cliente. A sequência de valores determina a ordem de serviço dos clientes, e o conjunto de clientes que compõem cada rota é limitado pela capacidade

dos veículos, ou seja, cada cliente é atribuído a um veículo específico na sequência em que aparece no vetor  $e$ , quando a capacidade do veículo é ultrapassada, uma nova rota é iniciada.

Tendo em vista que o PRV é uma generalização do Problema do Caixeiro Viajante (PCV) [Flood 1956], alguns autores realizam estudos comparativos aplicando métodos de cruzamento geralmente utilizados no PCV ao PRVC [Haj-Rachid et al. 2010, Nazif and Lee 2012, Puljić and Manger 2013, Varun Kumar and Panneerselvam 2017]. Em [Ahmed et al. 2023], os autores realizam um estudo comparativo entre múltiplos métodos encontrados na literatura do PCV, sendo eles, métodos específicos de cruzamento que levam em conta a distância entre os clientes e métodos que não levam em conta quaisquer dados relacionados a instância do problema.

Além dos operadores de cruzamento clássicos apresentados anteriormente, há também operadores de cruzamento utilizados em outras variantes do PRV que podem ser facilmente adaptados ao PRVC. No trabalho de Varun e outros autores [Varun Kumar and Panneerselvam 2017], propõem um novo operador de cruzamento, *Sinusoidal Motion Crossover Operator* (SMC), inspirado no movimento senoidal [Serway and Jewett Jr 2023]. Conforme os autores, além de gerar dois descendentes simultaneamente, o SMC, garante maior aleatoriedade nos descendentes gerados.

### 3. Referencial Teórico

O objetivo desta seção é apresentar a formulação inicial do PRV e suas variações (Seção 3.1), explicar o funcionamento dos algoritmos genéticos (Seção 3.2) e seus principais conceitos: Seleção (Seção 3.2.1.1), Cruzamento (Seção 3.2.1.2), Mutação (Seção 3.2.1.3) e seus parâmetros (Seção 3.2.2).

#### 3.1. Problema do Roteamento de Veículos

Dantzig e Ramser [Dantzig and Ramser 1959] foram os primeiros a introduzir o “Problema do Despacho de Caminhões”, modelando como uma frota de caminhões homogêneos, que poderia atender a demanda de petróleo de vários postos de gasolina a partir de um depósito central e com uma distância mínima percorrida. Cinco anos depois, Clarke e Wright [Clarke and Wright 1964] generalizaram este problema para um problema de otimização linear comumente encontrado no domínio da logística e transporte, ou seja, como servir um conjunto de clientes, geograficamente disperso ao redor do depósito central, utilizando uma frota de caminhões com capacidades variadas. Isso ficou conhecido como o Problema do Roteamento de Veículos (PRV), um dos tópicos mais estudados na área de Pesquisa Operacional [Braekers et al. 2016]. A modelagem utilizada foi proposta por Noon e outros autores [Noon et al. 1994], podemos defini-la da seguinte forma:

$$\min \sum_m \sum_{ij} d_{ij} y_{mij}. \quad (1)$$

$$\sum_m \sum_{i, i \neq j} y_{mij} + \sum_m \sum_{k, j \neq k} y_{mjk} = 2, \text{ para todo } j. \quad (2)$$

$$\sum_j y_{m0j} = 1, \text{ para todo } m. \quad (3)$$

$$\sum_{i, i \neq j} y_{mij} \leq 1, \text{ para todo } m, j. \quad (4)$$

$$\sum_i y_{mi0} = 1, \text{ para todo } m. \quad (5)$$

$$\sum_i y_{mij} - \sum_k y_{mjk} = 0, \text{ para todo } m, j. \quad (6)$$

$$\sum_i \sum_j y_{mij} \leq |S| - 1, \text{ para todo subconjunto } S, \text{ para todo } m. \quad (7)$$

$$\sum_{i,j} y_{mij} w_j \leq w, \text{ para todo } m. \quad (8)$$

O PRV abordado neste artigo, também é conhecido como Problema do Roteamento de Veículos Capacitado (PRVC), consiste em um único depósito,  $d$ , com  $n$  clientes e  $m$  veículos. Para cada cliente, o veículo deve pegar uma certa quantidade de peso,  $w_j$ , onde  $j$  é o número do cliente. O objetivo da Equação (1) é minimizar a distância total percorrida por todos os veículos onde  $d_{ij}$  é a distância do cliente  $i$  ao cliente  $j$ . A variável binária  $y_{mij}$  será igual a 1 se o veículo  $m$  for do cliente  $i$  para o cliente  $j$  e 0 caso contrário. A restrição imposta pela Equação (2) garante que exatamente um veículo visite e saia de cada cliente. Por meio da Equação (3), cada veículo é forçado a deixar o depósito. A Equação (4) garante que um veículo não visite um determinado cliente mais de uma vez e a Equação (5) garante que o veículo retorne ao depósito. A conservação de fluxo para cada passeio de veículo é imposta pela Equação (6). Sub-rotas são eliminadas pela Equação (7). A restrição de capacidade imposta pela Equação (8) garante que a quantidade total de peso levada pelo veículo não exceda um limite de peso de  $w$ .

Breakers e outros autores [Braekers et al. 2016] mostram que existem inúmeras características, cujas combinações geram variantes do PRV, dentre essas variantes, destacam-se [Braekers et al. 2016]:

- O Problema do roteamento de Veículos com Frota Heterogênea (PRVFH) resulta da combinação das características do PRV com uma frota de veículos heterogênea [Golden et al. 1984]. Neste problema, um único centro de distribuição é responsável por atender  $n$  clientes por dia com um único produto, usando uma frota de veículos heterogênea que pode ser limitada ou ilimitada. Os veículos têm restrição de capacidade  $w$  e a demanda de cada cliente  $i$  é  $d_i$ . Eles saem e retornam ao depósito  $d$ . As principais formulações e desigualdades do problema são apresentadas por Yaman [Yaman 2006].

- O Problema do Roteamento de Veículos com Múltiplos Depósitos (PRVMD) [Montoya-Torres et al. 2015], o objetivo é encontrar a melhor maneira de distribuir os produtos eficientemente para os clientes, considerando as variáveis e restrições do problema de localização e roteamento. Entre as restrições do problema estão as capacidades limitadas dos veículos, o tempo de trabalho dos motoristas, a capacidade de cada depósito e a possibilidade de pontos de vendas serem atendidos por mais de um veículo. Além disso, outros fatores podem influenciar a solução do problema, como as condições das estradas e o tráfego nas ruas. Para resolver o PRVMD, é necessário encontrar as rotas com o menor custo possível, considerando todas essas restrições e fatores. Para isso, são utilizadas técnicas de otimização e modelagem matemática, que permitem a exploração de diferentes cenários e encontrar a melhor solução possível.
- O Problema do Roteamento de Veículos com Janela de Tempo (PRVJT) [Bräysy and Gendreau 2005] possui a restrição adicional de que cada cliente  $i$  possui um tempo fixo de atendimento e uma janela de tempo  $[a_i, b_i]$  com extremos que representam, respectivamente, a abertura e fechamento da janela de tempo do cliente  $i$ , onde  $a_i$  é o horário inicial e  $b_i$  o tempo limite em que se pode realizar a entrega. Caso o veículo chegue antes do início da janela, este deve esperar, caso chegue depois da janela, o cliente é atendido sendo acrescida uma multa à empresa que realiza a entrega. O tempo de espera corresponde ao tempo em que o veículo chega a um cliente até o momento em que este deve começar ser atendido.

As variantes do PRV não se limitam apenas as citadas acima, tratam-se das mais pesquisadas no estado da arte, outras variantes podem ser encontradas no trabalho de Braekers e outros autores [Braekers et al. 2016].

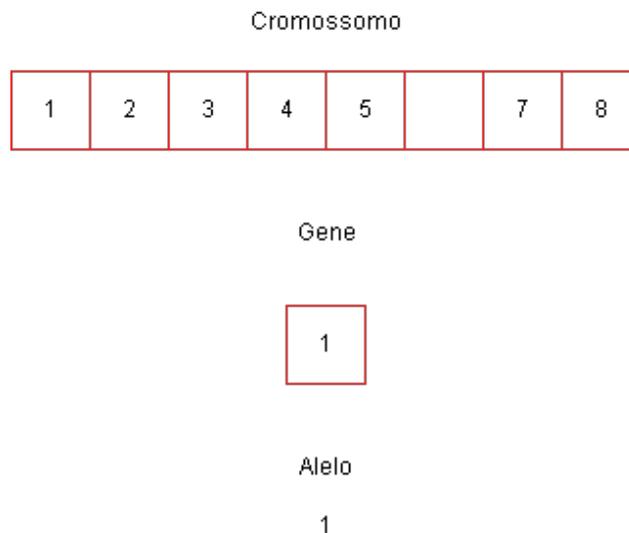
### 3.2. Algoritmos Genéticos

Os AGs são modelos computacionais de busca e otimização que se utilizam de técnicas para gerar e melhorar possíveis soluções candidatas a fim de encontrar uma solução próxima à ótima ou ótima para um determinado problema. Essas técnicas são baseadas no princípio de seleção natural e reprodução genética de Darwin [Darwin 1859].

Um cromossomo em AG representa uma solução possível do problema codificada em uma combinação de genes. Por sua vez, genes são símbolos que constituem os cromossomos e os valores possíveis que um gene pode assumir, são chamados de alelos [Gabriel and Delbem 2008]. Em outras palavras, podemos representar os conceitos abordados nesse parágrafo por meio da Figura 1.

De maneira geral, podemos definir um algoritmo genético no seguinte passo-a-passo [Silva Melo and Ferreira Filho 2001]: começamos com uma população aleatória de indivíduos, avaliados com base em seus cromossomos e associados a um valor que indica a aptidão de cada um, conforme o problema.

Após calcular a aptidão, avaliamos se algum indivíduo da população alcançou a solução ótima desejada. Caso contrário, precisamos continuar procurando no espaço de busca. Com a população atual, podemos manter inalterada uma parcela dos melhores indivíduos, este conceito é chamado de elitismo, com a parcela restante da população



**Figura 1. Exemplo de Cromossomo, Gene e Alelo.**

podemos alterá-los por meio de um operador de mutação, que altera os cromossomos de um indivíduo, podemos também, combinar dois indivíduos, em um novo, que combina suas respectivas características, através da reprodução.

Os indivíduos que vão compor essas operações podem ser selecionados por algum método de seleção. Estes passos são aplicados a cada geração a fim de se alcançar resultados cada vez mais próximos à solução ótima.

Equivalentemente, podemos também descrevê-lo em forma de pseudocódigo, como mostrado no Algoritmo 1:

---

**Algoritmo 1: ALGORITMO GENÉTICO CLÁSSICO**

---

**Entrada:** Alguns parâmetros como: número de gerações, taxa de mutação, de reprodução, etc..

**Saída:** Solução otimizada, estabelecida de acordo com os critérios do problema.

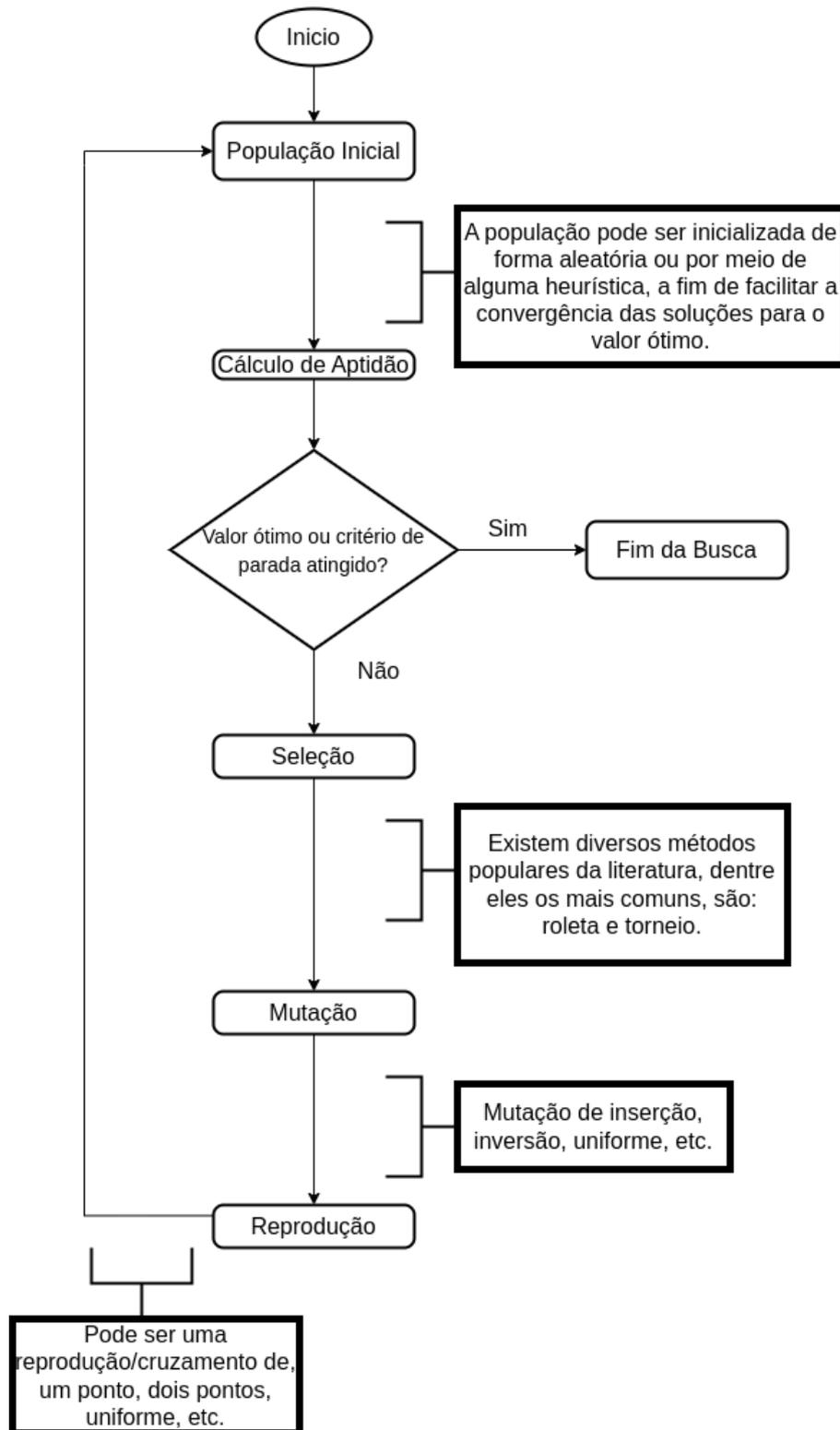
```

1 início
2   repita
3     sorteiaPais();
4     reproducaoIndividuos();
5     mutacaoIndividuos();
6     substituiNaPopulacao();
7   até resultadoOtimo() OU criterioParada();
8 fim
9 retorna POPULAÇÃO DE INDIVÍDUOS VÁLIDOS

```

---

A estrutura básica de um AG pode ser ilustrada por meio da Figura 2:



**Figura 2. Fluxograma de um AG básico.**

A seguir, falaremos aprofundadamente sobre os passos relatados na Figura 2.

### 3.2.1. Conceitos básicos de AGs

#### 3.2.1.1. Seleção

A etapa de seleção em um AG determina quais membros da população participarão da etapa de cruzamento/reprodução, o mesmo também pode ser chamado de operador de reprodução [Katoch et al. 2021].

- O método de seleção por roleta [Eiben and Smith 2015], mapeia cada solução com uma porção de uma roleta, onde essa porção é definida pela aptidão da solução, essa roleta é então girada para selecionar as soluções que participarão da reprodução para a próxima geração.
- A técnica de seleção por torneio foi proposta pela primeira vez por Brindle [Brindle 1980]. Nesse método, vários indivíduos da população são escolhidos aleatoriamente, e o indivíduo com a melhor aptidão entre eles é selecionado para reprodução. Esse processo é repetido até atingir o número desejado de indivíduos selecionados. A escolha do tamanho do torneio pode afetar o desempenho do algoritmo genético. Torneios muito pequenos podem levar a uma exploração excessiva, enquanto torneios muito grandes podem reduzir a pressão seletiva [Brindle 1980].

#### 3.2.1.2. Cruzamento

O operador de cruzamento é utilizado para combinar características de dois indivíduos ou mais, chamados de pais, gerando assim, um ou mais filhos, por isso, também é chamado de operador de reprodução. Existem diversos operadores de cruzamento, dentre eles: um ponto, dois pontos, uniforme, etc.

No cruzamento de um ponto [Katoch et al. 2021], tomam-se dois indivíduos (cromossomos), então, defini-se aleatoriamente um ponto de corte que particionará ambos os cromossomos em duas partes. Posteriormente, a primeira parte do primeiro cromossomo será agrupada à segunda parte do segundo cromossomo, formando, assim, o primeiro filho. Depois, a primeira parte do segundo cromossomo será agrupada à segunda parte do primeiro cromossomo formando o segundo filho. Também é mostrado através da Figura 3.

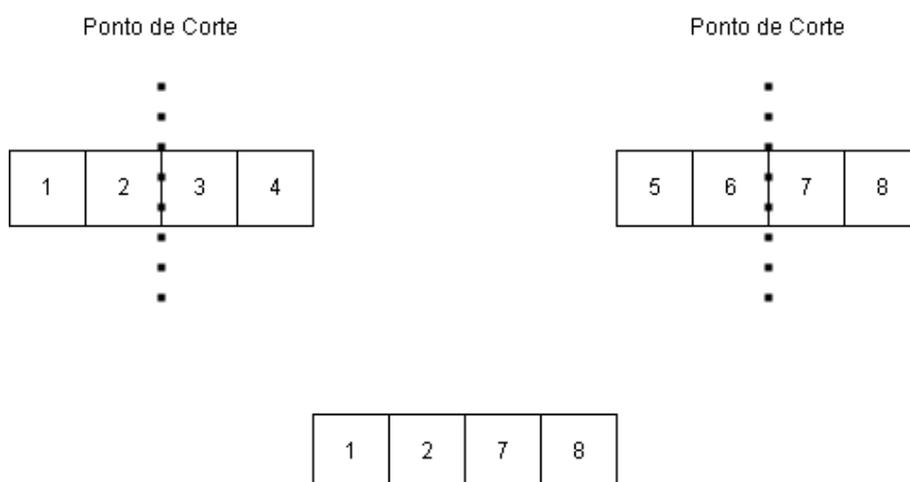
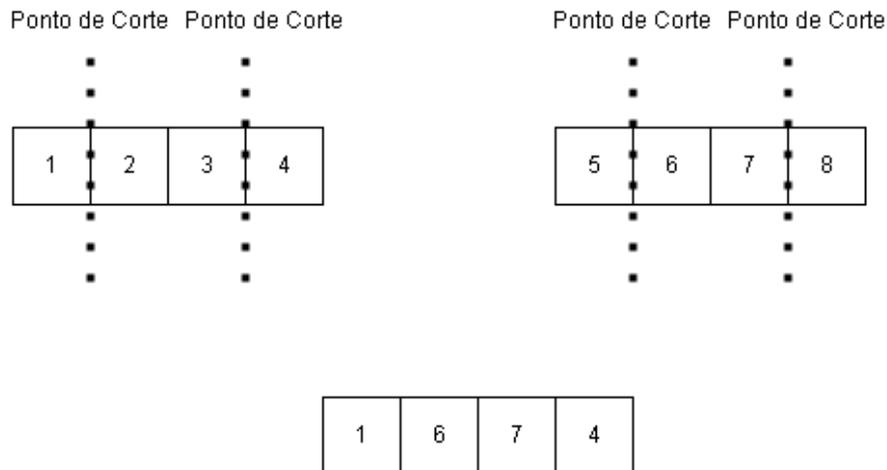


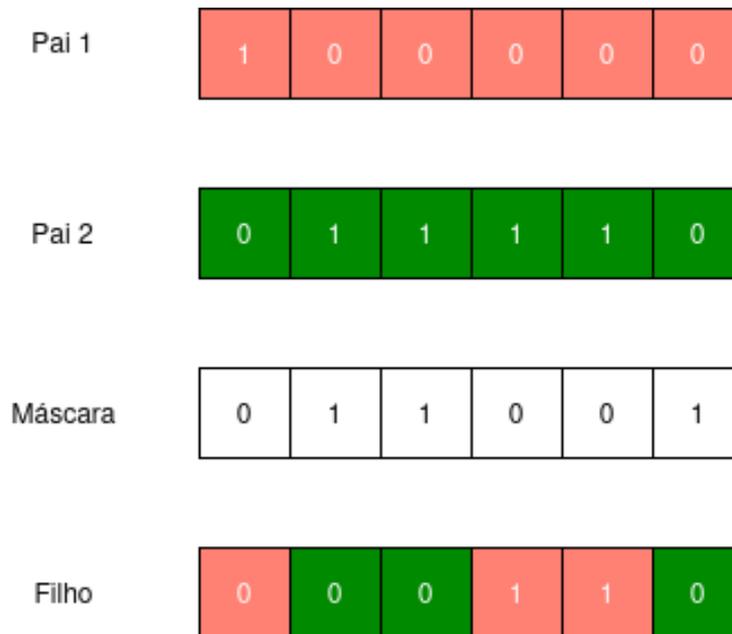
Figura 3. Exemplo de Cruzamento de um ponto de um dos filhos.

Já no cruzamento de dois pontos [Katoch et al. 2021], são definidos dois pontos, onde se copiam os genes do primeiro pai até o primeiro ponto de corte, para o filho. Em seguida, os genes do segundo pai, do primeiro ponto ao segundo ponto, são copiados para o novo indivíduo e, por fim, os genes do segundo ponto até o fim do cromossomo do primeiro pai são copiados para o novo indivíduo, como mostrado na Figura 4.



**Figura 4. Exemplo de Cruzamento de dois pontos de um dos filhos.**

No cruzamento uniforme [Chaudhry and Usman 2017], uma máscara aleatória de 0 e 1 é gerada conforme a taxa de cruzamento. Para uma taxa de cruzamento de 50%, metade dos genes nos filhos seria herdada do pai 1, enquanto a outra metade seria herdada do pai 2. Os genes que correspondem ao *bit* 1 são retirados do pai 1, enquanto os correspondentes a 0 são retirados do pai 2, como mostrado na Figura 5.

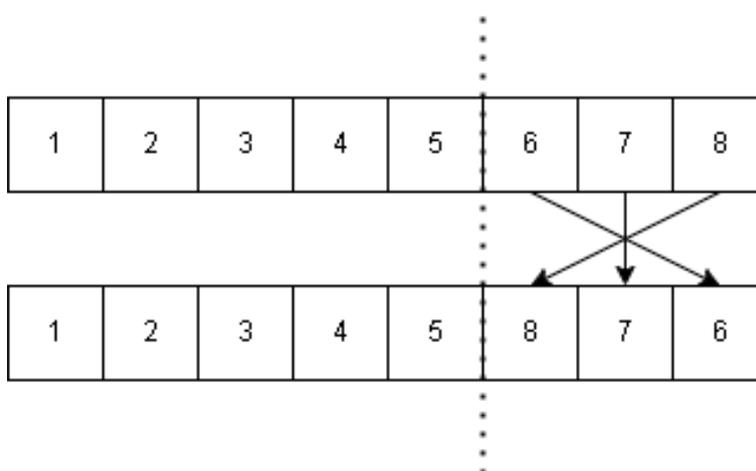


**Figura 5. Exemplo de cruzamento uniforme de um dos filhos.**

### 3.2.1.3. Mutação

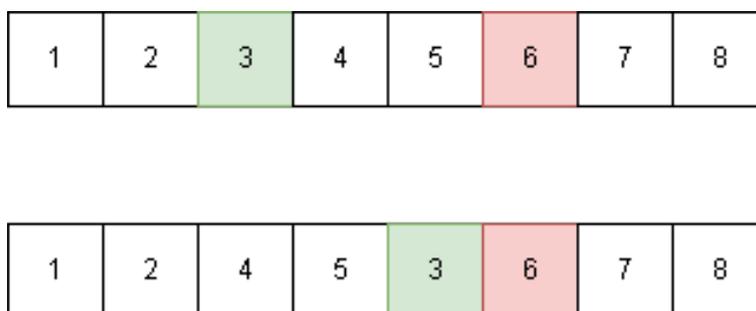
O operador de mutação altera arbitrariamente um ou mais alelos do cromossomo, fornecendo assim, meios para alterar um indivíduo [Nitasha and Kumar 2014]. A mutação é uma operação que utiliza apenas o indivíduo pai para criar o indivíduo filho, aplicando algum tipo de modificação aleatória em sua representação. Diversos tipos de mutação são descritos por Nitasha e Kumar [Nitasha and Kumar 2014], tais como: a mutação de inversão simples, inserção e uniforme.

O operador de Mutação de Inversão Simples (MIS) [Nitasha and Kumar 2014] reverte a subcadeia entre quaisquer dois locais especificados em uma solução individual. MIS é um operador de inversão que inverte a cadeia selecionada aleatoriamente e a coloca em um local aleatório, como mostrado na Figura 6.



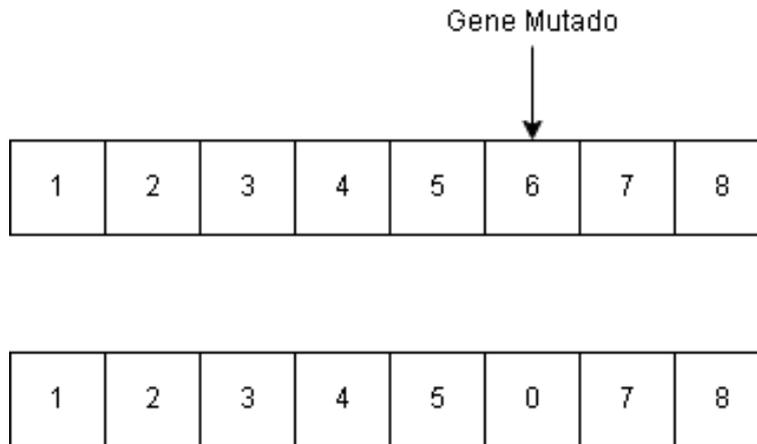
**Figura 6. Exemplo de Mutação MIS.**

Na mutação de inserção [Nitasha and Kumar 2014], seleciona-se dois genes e move-se o primeiro gene, para próximo do segundo gene, e com isso mudam-se as posições de todos os outros genes no processo.



**Figura 7. Exemplo de Mutação de Inserção.**

A mutação uniforme [Nitasha and Kumar 2014] consiste na alteração de um gene aleatório com base em um valor específico que pode ser assumido pelo gene. Em outras palavras, um gene selecionado sofre uma mutação que resulta em um novo valor, que é um dos valores possíveis permitidos pelo problema.



**Figura 8. Exemplo de Mutação Uniforme.**

### 3.2.2. Parâmetros

Dentre os diversos parâmetros [Eiben and Smith 2015, Gabriel and Delbem 2008] que podem ser ajustados nos AGs para obter um melhor resultado, destacam-se:

- **Gerações**  
Em um AG pode ser usado o número de gerações como critério de parada para a execução do algoritmo. Um pequeno número de gerações pode não produzir uma solução satisfatória para o problema, por outro lado, um número grande de gerações pode gastar um tempo computacional inviável para gerar uma solução satisfatória.
- **Tamanho da População**  
Um parâmetro que pode ser estático, permanecendo o mesmo durante toda a execução do programa, ou pode ser adaptativo, possuindo algum critério para alterar o tamanho da população e, conseqüentemente, cobrindo um espaço maior em busca das soluções.
- **Probabilidade de Cruzamento**  
Indica a chance de um indivíduo da população trocar material genético com outro da mesma população, assim, gerando um novo indivíduo que combina características de seus predecessores.
- **Probabilidade de Mutação**  
Dita a probabilidade de um indivíduo ter suas características alteradas. É responsável por evitar que as soluções do AG fiquem presas em mínimos locais.

## 4. Abordagem Proposta para Solução do PRVC

O algoritmo implementado neste trabalho utiliza o conceito de chaves aleatórias para codificar a solução nos cromossomos. A reprodução é feita por meio do cruzamento

uniforme, a mutação através da mutação por imigração e a seleção, por meio do método de torneio.

#### 4.1. Avaliação de Cromossomos

Para avaliar a aptidão de um cromossomo, leva-se em consideração a distância euclidiana, a distância entre pontos  $n$ -dimensionais,  $P = (P_1, P_2, \dots, P_n)$  e  $Q = (Q_1, Q_2, \dots, Q_n)$ , é obtida através da Equação (9):

$$\sqrt{(P_1 - Q_1)^2 + (P_2 - Q_2)^2 + \dots + (P_N - Q_N)^2} \quad (9)$$

O algoritmo permitirá rotas inválidas, ou seja, permitirá que as soluções infrinjam as restrições impostas na Seção 3.1. Quando uma solução infringir as restrições impostas, ela será penalizada. A aptidão ( $Apt$ ) do cromossomo será a distância total percorrida  $Dt$ , acrescida da raiz quadrada da soma de diferenças entre a capacidade do caminhão e o quanto a mesma foi excedida, multiplicado pelo número de caminhões que excederam sua capacidade,  $Ce$  [Wester 1993].

$$Apt(c) = Dt + \sqrt{\sum_{i=1}^n \max(0, P_i - C_i) \cdot Ce} \quad (10)$$

#### 4.2. Codificação da Solução

As chaves aleatórias, introduzidas por Bean [Bean 1993], têm como principal objetivo abordar o problema de geração de soluções inviáveis durante a reprodução, o qual surge quando alguns clientes são visitados repetidamente enquanto outros não são visitados [Bean 1993].

Considere um veículo ( $k = 1$ ) e seis clientes ( $c = 6$ ), podemos ter os seguintes cromossomos da Figura 9:



**Figura 9. Exemplo de cromossomos utilizando uma representação inteira.**

Utilizando a representação inteira, cada alelo contém um número inteiro que representa o número do cliente a ser visitado. Utilizando o cruzamento uniforme (Seção 3.2.1.2), temos o cromossomo resultante mostrado na Figura 10.

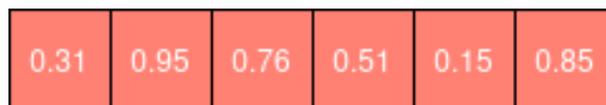


**Figura 10. Exemplo de cromossomos utilizando uma representação inteira.**

Após a reprodução, o novo indivíduo visitou o cliente 5 duas vezes e o cliente 4 nem ao menos foi visitado.

Com as chaves aleatórias podemos evitar esse tipo de reprodução inadequada. Alterando o tipo de representação do cromossomo para ponto flutuante, utilizamos as chaves aleatórias, geramos um número entre 0 e 1 para cada cliente, os clientes são representados por meio do índice de sua posição e a ordem de visitas aos clientes é representada pela ordenação das chaves aleatórias em ordem crescente.

A Figura 11 elucida o uso das chaves aleatórias para representar um cromossomo.

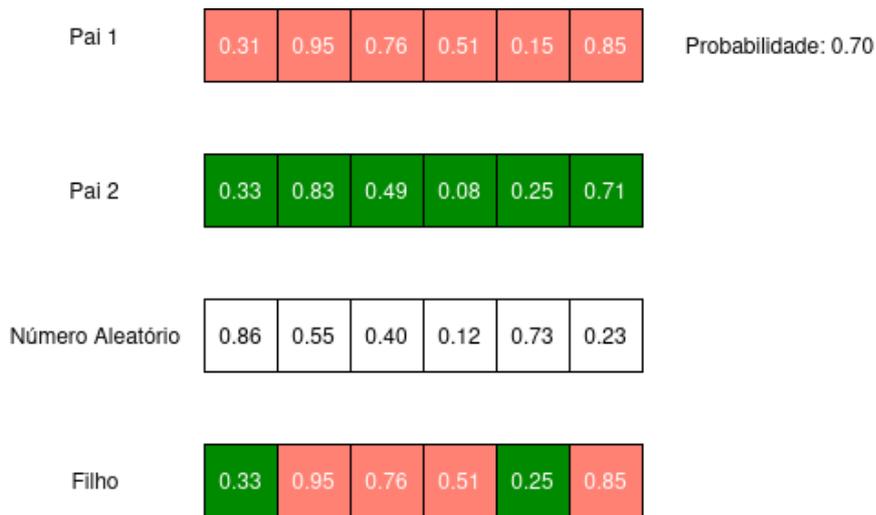


**Figura 11. Exemplo de representação de cromossomo utilizando chaves aleatórias.**

### 4.3. Método de Cruzamento

O método de cruzamento utilizado é descrito no trabalho de Bean [Bean 1993], um método específico para ser usado em conjunto com as chaves aleatórias. Nesse método, um número aleatório é gerado para cada gene. Se o número aleatório for maior que a probabilidade atribuída ao pai número 1, o gene é retirado do pai número 2; caso contrário, o gene é retirado do pai número 1 [Bean 1993, Wester 1993].

A Figura 12 mostra o comportamento do cruzamento aplicado nas chaves aleatórias.



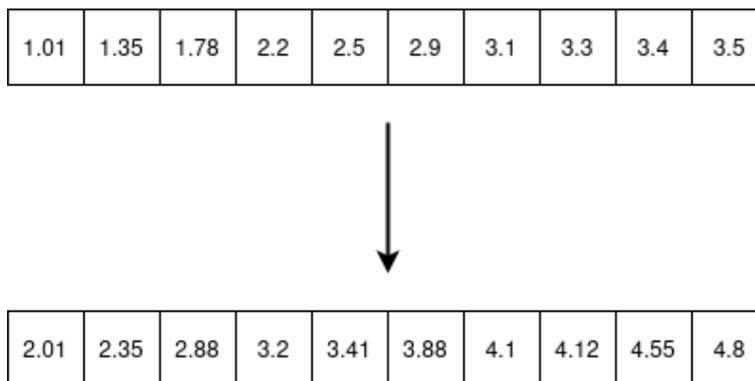
**Figura 12. Exemplo de cruzamento utilizando chaves aleatórias.**

Para suportar mais de um veículo, basta utilizar valores entre 1 e  $k + 1$ .

#### 4.4. Método de Mutação

A mutação por imigração [Bean 1993] consiste em alterar todo o cromossomo. De acordo com Bean [Bean 1993], essa mutação é utilizada para prevenir a convergência prematura da população utilizando as chaves aleatórias. Esse método não foi citado na seção de operadores de mutação (3.2.1.3) devido a não ser popular na literatura do PRV, os que foram citados são os mais comuns.

Aplicando a mutação por imigração utilizando a representação de chaves aleatórias temos o exemplo mostrado na Figura 13:



**Figura 13. Exemplo de mutação por imigração.**

#### 4.5. Método de Seleção

O método de seleção por torneio é outra técnica comumente utilizada em algoritmos genéticos para selecionar indivíduos para reprodução. Nesse método, um subconjunto de indivíduos da população é selecionado aleatoriamente para competir entre si, e o indivíduo mais apto desse grupo é escolhido como um dos pais para a próxima geração. Esse processo é repetido até que o número desejado de pais seja selecionado [Eiben and Smith 2015].

O algoritmo proposto utiliza um parâmetro denominado "tamanho do torneio" [Eiben and Smith 2015], que define quantos indivíduos participarão de uma competição. Nesse processo, dois indivíduos são escolhidos aleatoriamente da população e são utilizados para participarem da reprodução. Esse processo é repetido até que o tamanho especificado para o torneio seja alcançado.

Após a criação dos novos indivíduos, aqueles que apresentam um valor de aptidão menor, substituirão o indivíduo da população que tinha a menor aptidão entre os pais selecionados na etapa anterior do torneio.

## 5. Experimentos e Resultados

O algoritmo descrito neste trabalho foi implementado na linguagem C++ [Cplusplus 2012], e está disponível no repositório Github [Victor 2023], os experimentos foram realizados em um ambiente com a seguinte configuração: Processador 11th Gen Intel® Core™ i5-1135G7 @ 2.40GHz × 8; 20 GB de Memória RAM; Sistema Operacional Linux Ubuntu 22.04 LTS.

As instâncias que foram utilizadas nos testes, estão disponíveis na CVRPLIB [Lima et al. 2014] e pertencem aos conjuntos de dados: Conjunto A [Augerat 1995], Conjunto E [Christofides and Eilon 1969] e Conjunto P [Augerat 1995]. Para cada instância de teste foram realizadas 30 execuções. Como forma de avaliar a qualidade das soluções obtidas pelos algoritmos propostos, utilizamos o conceito de *gap*. O *gap* é calculado utilizando a Equação (11), é obtido pela média das soluções encontradas em comparação com o resultado ótimo disponibilizado na instância de teste.

$$gap(\%) = \frac{ResultadoObtido - ResultadoOtimo}{ResultadoOtimo} * 100 \quad (11)$$

Nas tabelas a seguir, os resultados são as distâncias percorridas pelos veículos<sup>1</sup> e o tempo está em segundos<sup>2</sup>.

Os resultados apresentados nas Tabelas 1, 2 e 3, mostram o comportamento do algoritmo, utilizando os conjuntos citados anteriormente. Os parâmetros utilizados na execução do experimento a seguir são: Probabilidade de Mutação: 40%, Probabilidade de Cruzamento: 80%, Tamanho da População: 10000, Número de Gerações: 1000 .

<b>Conjunto</b>	<b>Melhor Resultado<sup>1</sup></b>	<b>Resultado Médio<sup>1</sup></b>	<b>Tempo<sup>2</sup></b>	<b>Desvio Padrão</b>	<b>Valor Ótimo</b>	<b>GAP</b>
P-n16-k8	412,5300	489,9410	7,13	37,1973	450	8,88%
P-n23-k8	608,1500	675,8332	8,90	47,3906	529	27,76%
P-n22-k8	781,9100	893,2187	8,71	67,1171	603	48,13%
P-n19-k2	325,3800	338,8335	7,16	10,5309	212	59,83%
P-n20-k2	317,5900	350,6774	7,39	15,8557	216	62,35%
P-n21-k2	333,6000	354,4123	7,58	12,1163	211	67,97%
P-n22-k2	363,3600	371,9642	7,97	10,5583	216	72,21%
P-n55-k15	1.699,9600	1.803,5713	16,35	53,8993	989	82,36%
P-n60-k15	1.930,7600	2.033,0158	17,03	89,3426	968	110,02%
P-n51-k10	1.526,0500	1.712,2206	15,45	84,8734	741	131,07%
P-n50-k10	1.496,5800	1.636,0632	15,00	96,7625	696	135,07%
P-n40-k5	1.068,9400	1.083,1584	12,55	27,1824	458	136,50%
P-n50-k8	1.429,3700	1.578,3261	14,45	85,0975	631	150,13%
P-n55-k10	1.711,2600	1.784,3803	16,26	72,6489	694	157,12%
P-n55-k8	1.468,5800	1.512,3284	16,13	34,7714	588	157,20%
P-n45-k5	1.300,7400	1.329,2271	13,42	43,6791	510	160,63%
P-n60-k10	1.843,8600	2.007,7887	17,35	97,3694	744	169,86%
P-n50-k7	1.380,6500	1.496,0894	14,74	72,1758	554	170,05%
P-n65-k10	2.071,8800	2.172,5858	18,26	100,0469	792	174,32%
P-n55-k7	1.502,4800	1.565,7052	15,68	39,2596	568	175,65%
P-n70-k10	2.127,2100	2.346,9868	20,03	113,5564	827	183,80%
P-n76-k4	2.199,0300	2.275,8894	21,16	71,0770	593	283,79%
P-n76-k5	2.402,1500	2.493,7790	20,97	85,1010	627	297,73%
P-n101-k4	2.925,8700	2.927,5432	27,35	50,0443	681	329,89%

**Tabela 1. Tabela de execução com a 1ª configuração contendo o conjunto P.**

Conjunto	Melhor Resultado <sup>1</sup>	Resultado Médio <sup>1</sup>	Tempo <sup>2</sup>	Desvio Padrão	Valor Ótimo	GAP
E-n33-k4	1.267,03	1.401,80	11,65	53,74	835,00	67,88%
E-n22-k4	633,65	671,02	9,71	34,01	375,00	78,94%
E-n23-k3	1.057,08	1.105,65	9,29	46,56	569,00	94,31%
E-n30-k3	1.141,45	1.265,33	12,74	58,85	534,00	136,95%
E-n76-k14	2.516,70	2.633,33	24,87	115,98	1.021,00	157,92%
E-n76-k10	2.475,40	2.581,46	21,45	92,73	830,00	211,02%
E-n51-k5	1.525,44	1.639,75	14,42	89,75	521,00	214,73%
E-n101-k14	3.457,13	3.513,65	31,61	103,18	1.071,00	228,07%
E-n76-k7	2.270,60	2.294,27	22,84	45,74	682,00	236,40%
E-n76-k8	2.430,29	2.553,64	23,42	112,34	735,00	247,43%
E-n101-k8	3.275,32	3.324,98	32,94	107,62	817,00	306,97%

**Tabela 2. Tabela de execução com a 1ª configuração contendo o conjunto E.**

Conjunto	Melhor Resultado <sup>1</sup>	Resultado Médio <sup>1</sup>	Tempo <sup>2</sup>	Desvio Padrão	Valor Ótimo	GAP
A-n32-k5	1.538,9500	1.621,8287	13,03	56,3657	784	106,87%
A-n33-k6	1.467,1500	1.548,5003	10,77	80,1614	742	108,69%
A-n34-k5	1.546,6400	1.632,0452	11,03	60,2191	778	109,77%
A-n36-k5	1.548,4800	1.687,6390	11,97	70,6855	799	111,22%
A-n45-k7	2.333,5700	2.437,2123	16,71	123,8828	1.146	112,67%
A-n33-k5	1.377,7700	1.446,6255	12,74	51,5853	661	118,85%
A-n37-k6	1.974,0500	2.126,0203	16,06	114,4177	949	124,03%
A-n37-k5	1.402,8100	1.515,7139	14,61	49,4081	669	126,56%
A-n39-k5	1.787,2100	1.933,4110	14,74	99,3549	822	135,21%
A-n46-k7	2.063,7300	2.194,6926	15,97	78,6008	914	140,12%
A-n39-k6	1.846,8600	2.000,2077	12,90	83,2071	831	140,70%
A-n48-k7	2.566,5800	2.629,2400	16,74	102,4647	1.073	145,04%
A-n63-k9	3.780,6400	3.980,2226	17,61	212,1362	1.616	146,30%
A-n44-k6	2.072,3900	2.345,9148	17,32	155,9460	937	150,36%
A-n64-k9	3.251,3700	3.552,5523	22,10	204,2276	1.401	153,57%
A-n60-k9	3.256,2400	3.493,2926	23,32	174,4447	1.354	158,00%
A-n63-k10	3.229,1200	3.475,3239	22,68	169,2964	1.314	164,48%
A-n38-k5	1.770,6400	1.945,7829	15,52	108,9753	730	166,55%
A-n54-k7	2.752,6700	3.126,5087	18,81	198,0553	1.167	167,91%
A-n80-k10	4.470,3700	4.771,7097	27,06	211,8796	1.763	170,66%
A-n55-k9	2.801,4200	2.976,4029	20,42	142,6614	1.073	177,39%
A-n62-k8	3.423,7900	3.671,8632	22,42	169,2052	1.288	185,08%
A-n45-k6	2.542,1100	2.698,1003	15,16	156,5347	944	185,82%
A-n61-k9	2.774,6300	2.977,7442	22,39	129,2202	1.034	187,98%
A-n53-k7	2.726,8400	2.953,1174	21,58	165,7844	1.010	192,39%
A-n65-k9	3.447,2200	3.724,5235	22,52	178,6666	1.174	217,25%
A-n69-k9	3.512,2300	3.744,9329	25,52	165,3921	1.159	223,12%

**Tabela 3. Tabela de execução com a 1ª configuração contendo o conjunto A.**

Usualmente, são feitos experimentos ajustando os parâmetros do AG até que se obtenham resultados satisfatórios. Na segunda configuração dos experimentos, foram utilizados parâmetros encontrados em outros trabalhos. O tamanho da população foi definido como 105 [Wang and Lu 2009, Baker and Ayechev 2003]. O número de gerações, 100.000 [Baker and Ayechev 2003]. A probabilidade de mutação, cerca de 5% [Schaffer et al. 1989], e a probabilidade de cruzamento 80% [Schaffer et al. 1989].

<b>Conjunto</b>	<b>Melhor Resultado<sup>1</sup></b>	<b>Resultado Médio<sup>1</sup></b>	<b>Tempo<sup>2</sup></b>	<b>Desvio Padrão</b>	<b>Valor Ótimo</b>	<b>GAP</b>
P-n16-k8	448.25	487,32	17,26	30,39	450	8,29%
P-n23-k8	617.8	667,78	23,55	36,14	529	26,24%
P-n22-k8	755.89	863,36	21,61	63,17	603	43,18%
P-n19-k2	328.77	376,22	19,77	18,43	212	77,46%
P-n55-k15	1718.71	1.807,77	32,13	65,45	989	82,79%
P-n20-k2	372.57	396,15	21,10	20,81	216	83,40%
P-n21-k2	375.06	399,59	23,16	17,71	211	89,38%
P-n22-k2	401.15	423,80	21,13	18,28	216	96,20%
P-n60-k15	1969.94	2.081,62	33,68	64,17	968	115,04%
P-n50-k10	1564.78	1.628,44	33,48	60,01	696	133,97%
P-n51-k10	1670.72	1.752,34	32,90	67,00	741	136,48%
P-n55-k10	1571.87	1.733,97	32,45	72,51	694	149,85%
P-n50-k8	1489.89	1.593,29	33,55	67,92	631	152,50%
P-n40-k5	1145.14	1.199,67	27,00	45,18	458	161,94%
P-n60-k10	1878.25	1.999,66	32,48	86,51	744	168,77%
P-n55-k8	1571.15	1.629,76	30,13	59,72	588	177,17%
P-n65-k10	2019.67	2.211,59	35,55	102,01	792	179,24%
P-n50-k7	1441.84	1.569,87	33,03	82,00	554	183,37%
P-n70-k10	2273	2.397,74	37,00	110,07	827	189,93%
P-n45-k5	1367.35	1.499,42	26,65	81,70	510	194,00%
P-n55-k7	1551.12	1.684,28	31,13	85,05	568	196,53%
P-n76-k5	2397.43	2.484,00	41,35	94,85	627	296,17%
P-n76-k4	2267.19	2.474,52	41,26	121,30	593	317,29%
P-n101-k4	2947.72	3.048,64	73,00	66,58	681	347,67%

**Tabela 4. Tabela de execução com a 2ª configuração contendo o conjunto P.**

<b>Conjunto</b>	<b>Melhor Resultado<sup>1</sup></b>	<b>Resultado Médio<sup>1</sup></b>	<b>Tempo<sup>2</sup></b>	<b>Desvio Padrão</b>	<b>Valor Ótimo</b>	<b>GAP</b>
E-n33-k4	1.444,29	1.528,67	27,39	58,74	835,00	83,07%
E-n22-k4	610,25	791,12	18,94	86,64	375,00	110,97%
E-n23-k3	1.106,66	1.302,53	19,90	116,23	569,00	128,92%
E-n76-k14	2.508,55	2.618,09	50,03	102,05	1.021,00	156,42%
E-n30-k3	1.318,21	1.570,51	22,87	148,53	534,00	194,10%
E-n76-k10	2.425,05	2.553,05	50,68	105,90	830,00	207,60%
E-n51-k5	1.513,91	1.667,69	56,26	82,44	521,00	220,09%
E-n101-k14	3.460,91	3.545,35	53,55	82,79	1.071,00	231,03%
E-n76-k8	2.377,64	2.545,52	48,97	101,71	735,00	246,33%
E-n76-k7	2.254,55	2.442,60	49,03	115,07	682,00	258,15%
E-n101-k8	3.337,14	3.457,90	54,58	107,55	817,00	323,24%

**Tabela 5. Tabela de execução com a 2ª configuração contendo o conjunto E.**

Conjunto	Melhor Resultado <sup>1</sup>	Resultado Médio <sup>1</sup>	Tempo <sup>2</sup>	Desvio Padrão	Valor Ótimo	GAP
A-n45-k7	2.369,55	2.517,98	34,71	142,82	1.146,00	119,72%
A-n37-k6	1.882,14	2.104,54	46,48	129,56	949,00	121,76%
A-n33-k6	1.467,82	1.687,90	26,97	125,80	742,00	127,48%
A-n32-k5	1.701,10	1.816,51	22,55	83,15	784,00	131,70%
A-n36-k5	1.656,31	1.859,95	42,19	99,19	799,00	132,79%
A-n34-k5	1.634,53	1.825,32	35,71	156,71	778,00	134,62%
A-n37-k5	1.567,17	1.646,26	44,19	58,79	669,00	146,08%
A-n63-k9	3.836,43	4.033,53	53,74	229,87	1.616,00	149,60%
A-n33-k5	1.543,27	1.660,22	33,58	89,62	661,00	151,17%
A-n39-k5	1.784,93	2.075,49	44,35	133,97	822,00	152,49%
A-n64-k9	3.456,23	3.564,42	54,35	136,37	1.401,00	154,42%
A-n44-k6	2.158,89	2.391,19	39,13	110,87	937,00	155,20%
A-n60-k9	3.268,49	3.535,91	70,13	190,99	1.354,00	161,15%
A-n48-k7	2.489,12	2.803,06	36,97	169,32	1.073,00	161,24%
A-n63-k10	3.223,56	3.434,53	55,06	158,41	1.314,00	161,38%
A-n39-k6	1.954,01	2.199,77	36,52	136,70	831,00	164,71%
A-n54-k7	2.940,98	3.113,93	69,03	143,06	1.167,00	166,83%
A-n46-k7	2.247,01	2.451,02	30,42	158,16	914,00	168,16%
A-n80-k10	4.523,62	4.737,92	60,29	191,62	1.763,00	168,74%
A-n38-k5	1.792,43	1.963,33	41,32	121,89	730,00	168,95%
A-n55-k9	2.707,74	2.974,74	69,10	182,71	1.073,00	177,24%
A-n45-k6	2.416,41	2.686,24	40,48	156,56	944,00	184,56%
A-n62-k8	3.465,98	3.689,72	54,61	191,28	1.288,00	186,47%
A-n61-k9	2.739,20	2.999,67	58,32	137,41	1.034,00	190,10%
A-n53-k7	2.751,48	2.953,44	57,84	143,82	1.010,00	192,42%
A-n69-k9	3.367,19	3.665,77	52,94	182,71	1.159,00	216,29%
A-n65-k9	3.468,46	3.738,96	56,48	146,77	1.174,00	218,48%

**Tabela 6. Tabela de execução com a 2ª configuração contendo o conjunto A.**

O conjunto P apresentou uma melhora média de 7,12% no *gap* ao compararmos os resultados entre o 1ª e o 2ª experimento, conforme apresentado na Tabela 4. Em contraste, o conjunto A mostrou uma piora média de 6,26% no *gap* ao comparar os mesmos experimentos, conforme evidenciado na Tabela 6. Por fim, no conjunto E, observou-se uma piora média de 18,29% no *gap* ao comparar os resultados entre a primeira e a segunda configuração, conforme detalhado na Tabela 5.

Comparando os parâmetros de ambos os experimentos apresentados, por mais que a população dos experimentos apresentados nas Tabelas 1, 2, 3 seja maior que a população utilizada nos experimentos apresentados nas Tabelas 4, 5, 6, a variação no *gap* não foi significativa, muito provavelmente o número de gerações não é suficientemente grande para que o espaço de busca possa convergir.

Por outro lado, os experimentos apresentados nas Tabelas 4, 5, 6 possuem 100 vezes mais gerações que os experimentos relatados nas Tabelas 1, 2, 3, assim, não foi suficiente para diminuir significativamente o *gap*, possivelmente há uma relação entre o número de gerações e o tamanho da população. Por mais que o número de gerações seja 100 vezes maior, o tamanho da população é 10 vezes menor.

Em resumo, os dados levantados sugerem que simplesmente aumentar o número de gerações pode não ser suficiente para que o espaço de busca alcance uma convergência efetiva. Além disso, a diferença no tamanho das populações nos dois conjuntos de experi-

mentos parece desempenhar um papel crucial. Isso aponta para a importância de encontrar um equilíbrio adequado entre o número de gerações e o tamanho da população, um fator fundamental para garantir uma variabilidade genética eficaz. Essas descobertas levantam questões instigantes que influenciam nossa reflexão sobre como otimizar parâmetros em experimentos futuros, proporcionando uma oportunidade para uma compreensão mais profunda dos processos evolutivos que estão sendo implementados.

## 6. Conclusão e Trabalhos Futuros

Ao finalizar a análise dos experimentos, os resultados proporcionam um *insight* revelador sobre as dinâmicas e interações entre os conjuntos P, A e E. A avaliação crítica das variações nos *gaps* nos experimentos 1 e 2 revelou detalhes interessantes. O conjunto P apresentou uma melhoria média notável de 7,12%, enquanto o conjunto A teve uma média de piora de 6,26%, e o conjunto E surpreendentemente mostrou uma piora mais acentuada de 18,29%. Essa análise detalhada indica que a eficácia dos modelos varia consideravelmente em diferentes contextos, destacando a complexidade inerente à otimização de parâmetros em algoritmos evolutivos.

Além disso, ao considerar os resultados, fica evidente que alcançar o equilíbrio ideal entre o número de gerações e o tamanho da população é uma tarefa desafiadora. O aumento substancial nas gerações na segunda configuração, embora pudesse proporcionar uma variabilidade genética mais ampla, não foi suficiente para contrabalançar o impacto do menor tamanho da população. Este resultado reforça a relevância de abordagens mais abrangentes na otimização de algoritmos evolutivos.

Em conclusão, esta investigação ofereceu uma análise crítica das nuances da otimização de parâmetros em algoritmos evolutivos, apontando para áreas que demandam uma investigação mais aprofundada. Ao explorar as complexidades da relação entre o número de gerações e o tamanho da população, aspiramos a incentivar pesquisas futuras que aprimorem a compreensão desses processos evolutivos e suas implicações práticas. Este trabalho não só contribui para o corpo de conhecimento existente, mas também abre portas para a exploração de novas fronteiras na busca por eficiência e eficácia em algoritmos evolutivos aplicados.

Apesar dos resultados alcançados nesta pesquisa, há ainda alguns pontos a serem explorados:

- Explorar a interação entre o número de gerações e o tamanho da população em maior detalhe, considerando diferentes configurações e cenários. Investigar se há configuração ótima que maximize a variabilidade genética e minimize os *gaps*.
- Explorar a abordagem de não permitir que soluções inviáveis se mantenham na população de soluções e avaliar seu impacto, comparando-o com os resultados apresentados neste trabalho.
- Explorar a mudança de tipo de codificação dos cromossomos, trocando a abordagem de chaves aleatórias para uma mais difundida no estado da arte, como a representação inteira.
- Realizar experimentos para medir o impacto da troca de representação, comparando os novos resultados com os apresentados neste trabalho.

## Referências

- Ahmed, Z. H., Al-Otaibi, N., Al-Tameem, A., and Saudagar, A. K. J. (2023). Genetic crossover operators for the capacitated vehicle routing problem. *Comput. Mater. Contin.*, 74:1575–1605.
- Alfa, A. S., Heragu, S. S., and Chen, M. (1991). A 3-opt based simulated annealing algorithm for vehicle routing problems. *Computers & Industrial Engineering*, 21(1-4):635–639.
- Augerat, P. (1995). *Approche polyèdrale du problème de tournées de véhicules*. PhD thesis, Institut National Polytechnique de Grenoble-INPG.
- Baker, B. M. and Ayechev, M. (2003). A genetic algorithm for the vehicle routing problem. *Computers & Operations Research*, 30(5):787–800.
- Bean, J. C. (1993). Genetics and random keys for sequencing and optimization. Technical report.
- Braekers, K., Ramaekers, K., and Van Nieuwenhuysse, I. (2016). The vehicle routing problem: State of the art classification and review. *Computers & Industrial Engineering*, 99:300–313.
- Bräysy, O. and Gendreau, M. (2005). Vehicle routing problem with time windows, part i: Route construction and local search algorithms. *Transportation science*, 39(1):104–118.
- Brindle, A. (1980). *Genetic algorithms for function optimization*. PhD thesis, University of Alberta.
- Chaudhry, I. A. and Usman, M. (2017). Integrated process planning and scheduling using genetic algorithms. *Tehnički Vjesnik–Technical Gazette*, 24(5):1401–1409.
- Christofides, N. and Eilon, S. (1969). An algorithm for the vehicle-dispatching problem. *OR 1969-sep vol. 20 iss. 3*, 20.
- Clarke, G. and Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations research*, 12(4):568–581.
- Coloni, A., Dorigo, M., Maniezzo, V., et al. (1991). Distributed optimization by ant colonies. 142:134–142.
- Cplusplus, S. (2012). *ISO/IEC 14882:2011 Information technology — Programming languages — C++*. International Organization for Standardization, Geneva, Switzerland.
- Dantzig, G. B. and Ramser, J. H. (1959). The truck dispatching problem. *Management science*, 6(1):80–91.
- Darwin, C. (1859). *On the origin of species, 1859*. Routledge.
- de Araujo Lima, S. J. and de Araújo, S. A. (2020). Genetic algorithm applied to the capacitated vehicle routing problem: An analysis of the influence of different encoding schemes on the population behavior. *American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)*, 73(1):96–110.
- Dueck, G. and Scheuer, T. (1990). Threshold accepting: A general purpose optimization algorithm appearing superior to simulated annealing. *Journal of computational physics*, 90(1):161–175.

- Durbin, R. and Willshaw, D. (1987). An analogue approach to the travelling salesman problem using an elastic net method. *Nature*, 326(6114):689–691.
- Eiben, A. E. and Smith, J. E. (2015). *Introduction to evolutionary computing*. Springer.
- El Ghaziri, H. (1993). Algorithmes connexionnistes pour l’optimisation combinatoire. Technical report, EPFL.
- Fisher, M. L. and Jaikumar, R. (1981). A generalized assignment heuristic for vehicle routing. *Networks*, 11(2):109–124.
- Flood, M. M. (1956). The traveling-salesman problem. *Operations Research*, 4(1):61–75.
- Gabriel, P. H. R. and Delbem, A. C. B. (2008). Fundamentos de algoritmos evolutivos. Technical report.
- Gendreau, M., Hertz, A., and Laporte, G. (1994). A tabu search heuristic for the vehicle routing problem. *Management science*, 40(10):1276–1290.
- Gillett, B. E. and Miller, L. R. (1974). A heuristic algorithm for the vehicle-dispatch problem. *Operations Research*, 22(2):340–349.
- Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & operations research*, 13(5):533–549.
- Golden, B., Assad, A., Levy, L., and Gheysens, F. (1984). The fleet size and mix vehicle routing problem. *Computers & Operations Research*, 11(1):49–66.
- Haj-Rachid, M., Ramdane-Cherif, W., Chatonnay, P., and Bloch, C. (2010). Comparing the performance of genetic operators for the vehicle routing problem. *IFAC Proceedings Volumes*, 43(17):313–319.
- Katoch, S., Chauhan, S. S., and Kumar, V. (2021). A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications*, 80(5):8091–8126.
- Kawamura, H., Yamamoto, M., Mitamura, T., Suzuki, K., and Ohuchi, A. (1998). Cooperative search based on pheromone communication for vehicle routing problems. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, 81(6):1089–1096.
- Kirkpatrick, S., Gelatt Jr, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, 220(4598):671–680.
- Laporte, G. and Semet, F. (2002). Classical heuristics for the capacitated vrp. In *The vehicle routing problem*, pages 109–128. SIAM.
- Lima, I., Uchoa, and Oliveira, D. (2014). Cvrplib - capacitated vehicle routing problem library. <http://vrp.atd-lab.inf.puc-rio.br/index.php/en/>.
- Montoya-Torres, J. R., Franco, J. L., Isaza, S. N., Jiménez, H. F., and Herazo-Padilla, N. (2015). A literature review on the vehicle routing problem with multiple depots. *Computers & Industrial Engineering*, 79:115–129.
- Mulloorakam, A. T. and Nidhiry, N. M. (2019). Combined objective optimization for vehicle routing using genetic algorithm. *Materials Today: Proceedings*, 11:891–902.
- Nazif, H. and Lee, L. S. (2012). Optimised crossover genetic algorithm for capacitated vehicle routing problem. *Applied Mathematical Modelling*, 36(5):2110–2117.

- Nitasha, S. and Kumar, T. (2014). Study of various mutation operators in genetic algorithms. *International Journal of Computer Science and Information Technologies*, 5(3):4519–4521.
- Noon, C. E., Mittenthal, J., and Pillai, R. (1994). A tssp+ 1 decomposition strategy for the vehicle routing problem. *European journal of operational research*, 79(3):524–536.
- Osman, I. H. (1993). Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of operations research*, 41:421–451.
- Puljić, K. and Manger, R. (2013). Comparison of eight evolutionary crossover operators for the vehicle routing problem. *Mathematical Communications*, 18(2):359–375.
- Pureza, V. and França, P. M. (1991). Vehicle routing problems via tabu search metaheuristic. *Centre De Recherche Sur Les Transports Publication*, (747).
- Robust, F., Daganzo, C. F., and Souleyrette II, R. R. (1990). Implementing vehicle routing models. *Transportation Research Part B: Methodological*, 24(4):263–286.
- Schaffer, J. D., Caruana, R., Eshelman, L. J., and Das, R. (1989). A study of control parameters affecting online performance of genetic algorithms for function optimization. In *Proceedings of the 3rd international conference on genetic algorithms*, pages 51–60.
- Serway, R. A. and Jewett Jr, J. W. (2023). *Princípios de física-Vol. 3: Eletromagnetismo*, volume 3. Cengage Learning.
- Silva Melo, A. C. d. and Ferreira Filho, V. J. M. (2001). Sistemas de roteirização e programação de veículos. *Pesquisa operacional*, 21:223–232.
- Toth, P. and Vigo, D. (2002). The vehicle routing problem monographs on discrete mathematics and applications siam. *Philadelphia, Pennsylvania*.
- Varun Kumar, S. and Panneerselvam, R. (2017). A study of crossover operators for genetic algorithms to solve vrp and its variants and new sinusoidal motion crossover operator. *International Journal of Computational Intelligence Research*, 13(7):1717–1733.
- Victor, L. (2023). Uma implementação de algoritmo genético para o problema do roteamento de veículos capacitado (prvc). [https://github.com/victorlelissoares/Genetic\\_Algorithms](https://github.com/victorlelissoares/Genetic_Algorithms).
- Wang, C.-H. and Lu, J.-Z. (2009). A hybrid genetic algorithm that optimizes capacitated vehicle routing problems. *Expert Systems with Applications*, 36(2):2921–2936.
- Wester, V. D. (1993). *A genetic algorithm for the vehicle routing problem*. PhD thesis.
- Willard, J. (1989). Vehicle routing using r-optimal tabu search. *Master's thesis, The Management School, Imperial College, London*.
- Yaman, H. (2006). Formulations and valid inequalities for the heterogeneous vehicle routing problem. *Mathematical programming*, 106:365–390.