

UNIVERSIDADE FEDERAL DE OURO PRETO - UFOP
ESCOLA DE MINAS - EM
DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO,
ADMINISTRAÇÃO E ECONOMIA - DEPRO

CARLOS ANTÔNIO FELÍCIO TEODORO

Simulated Annealing e Algoritmos
Genéticos: Estudo Comparativo em
Problemas de Elevadas Dimensões

Ouro Preto
2021

Carlos Antônio Felício Teodoro

***Simulated Annealing* e Algoritmos Genéticos:
Estudo Comparativo em Problemas de Elevadas
Dimensões**

Monografia apresentada ao Curso de Engenharia de Produção da Universidade Federal de Ouro Preto como parte dos requisitos para a obtenção do Grau de Engenheiro de Produção.

Universidade Federal de Ouro Preto

Orientador: Prof. Dr. Helton Cristiano Gomes

Ouro Preto
2021

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

T314s Teodoro, Carlos Antônio Felício.
Simulated annealing e algoritmos genéticos [manuscrito]: estudo comparativo em problemas de elevadas dimensões. / Carlos Antônio Felício Teodoro. - 2021.
27 f.

Orientador: Prof. Dr. Helton Cristiano Gomes.
Monografia (Bacharelado). Universidade Federal de Ouro Preto.
Escola de Minas. Graduação em Engenharia de Produção .

1. Engenharia de produção. 2. Otimização matemática. 3. Algoritmos genéticos. I. Gomes, Helton Cristiano. II. Universidade Federal de Ouro Preto. III. Título.

CDU 658.5:519.6

Bibliotecário(a) Responsável: Sione Galvão Rodrigues - CRB6 / 2526



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
ESCOLA DE MINAS
DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO, ADMINISTRAÇÃO E ECON



FOLHA DE APROVAÇÃO

Carlos Antônio Felício Teodoro

Simulated Annealing e Algoritmos Genéticos: estudo comparativo em problemas de elevadas dimensões

Monografia apresentada ao curso de Engenharia de Produção da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Engenheiro de Produção.

Aprovada em 04 de outubro de 2021.

Membros da banca

Doutor - Helton Cristiano Gomes - Orientador - Universidade Federal de Ouro Preto
Mestre - Cristiano Luís Turbino de Franca e Silva - Universidade Federal de Ouro Preto
Mestrando - Ruan Carlos Silva Menezes Pinheiro - Universidade Federal de Ouro Preto
Mestrando - Pedro Henrique Silva - Universidade Federal de Minas Gerais
Mestrando - Davi Silva Alves - Universidade Federal de Minas Gerais

Helton Cristiano Gomes, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 11/10/2021.



Documento assinado eletronicamente por **Helton Cristiano Gomes, PROFESSOR DE MAGISTERIO SUPERIOR**, em 11/10/2021, às 14:41, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Cristiano Luis Turbino de Franca e Silva, PROFESSOR DE MAGISTERIO SUPERIOR**, em 11/10/2021, às 21:14, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0231613** e o código CRC **A88AA90C**.

Agradecimentos

Dedico este trabalho a minha família e a todos os que me ajudaram ao longo desta caminhada.

“Penso, logo existo.”
René Descartes

Resumo

Este estudo tem por objetivo realizar uma avaliação quantitativa acerca do emprego das meta-heurísticas *Simulated Annealing* (SA) e Algoritmos Genéticos (AG) para a resolução de problemas de roteamento de elevadas dimensões que, a princípio, não podem ser solucionados por métodos exatos em espaço de tempo razoável. A análise foi realizada a partir da implementação de cada um dos algoritmos, considerando diferentes cenários de parametrização, a fim de se comparar o desempenho e as soluções obtidas. Após a avaliação dos algoritmos, pôde-se concluir, para as instâncias utilizadas, que o SA foi capaz de encontrar melhores resultados em menor tempo de execução.

Palavras-chave: Algoritmos Genéticos, *Simulated Annealing*, roteamento, caixeiro viajante.

Abstract

This study aims to carry out a quantitative evaluation of the use of meta-heuristics Simulated Annealing (SA) and Genetic Algorithms (AG) to solve high-dimensional routing problems that, in principle, cannot be solved by combinatorial methods in a reasonable amount of time. The analysis was performed from the implementation of each of the algorithms, considering different parameterization scenarios, in order to compare performance and solutions. After evaluating the algorithms, it was possible to conclude, for the analyzed dataset, that SA was able to find the best results in a shorter execution time.

Keywords: genetic algorithm, simulated annealing, cashier traveler.

Lista de abreviaturas e siglas

| | |
|-----|-------------------------------|
| AG | Algoritmos Genéticos |
| SA | <i>Simulated Annealing</i> |
| PCV | Problema do Caixeiro Viajante |

Lista de ilustrações

| | |
|--|----|
| Figura 1 – Evolução do número de soluções de acordo com o aumento de cidades a serem visitadas no PCV. | 14 |
| Figura 2 – Exemplo matriz de distâncias. | 18 |
| Figura 3 – Operação de cruzamento de rotas. | 19 |
| Figura 4 – Evolução de execução. | 22 |
| Figura 5 – Evolução de execução para 100 localidades. | 24 |
| Figura 6 – Evolução de execução para 200 localidades. | 26 |

Sumário

| | | |
|-------|---|----|
| | Lista de ilustrações | 9 |
| 1 | INTRODUÇÃO | 11 |
| 1.1 | Objetivos | 11 |
| 1.1.1 | Objetivos Específicos | 12 |
| 1.1.2 | Justificativa | 12 |
| 2 | REFERENCIAL TEÓRICO | 13 |
| 2.1 | Logística e o Problema do Caixeiro Viajante | 13 |
| 2.2 | Meta-heurística <i>Simulated Annealing</i> | 14 |
| 2.3 | Algoritmo Genético | 16 |
| 3 | METODOLOGIA | 17 |
| 3.1 | Classificação da Pesquisa | 17 |
| 3.2 | Parâmetros | 17 |
| 3.3 | Gap | 17 |
| 3.4 | Matriz de distâncias | 17 |
| 3.5 | Implementação das meta-heurísticas | 18 |
| 3.6 | <i>Simulated Annealing</i> | 18 |
| 3.6.1 | Busca Local | 18 |
| 3.6.2 | Função de temperatura | 18 |
| 3.6.3 | Função de aceitação | 19 |
| 3.7 | Algoritmo Genético | 19 |
| 3.7.1 | Operação de cruzamento | 19 |
| 3.7.2 | Operação de mutação | 19 |
| 3.7.3 | Gerador de populações | 20 |
| 4 | APRESENTAÇÃO E DISCUSSÃO DOS RESULTADOS | 21 |
| 4.1 | Validação da Implementação | 21 |
| 4.2 | Cenário 01: 100 Localidades | 22 |
| 4.3 | Cenário 02: 200 Variáveis | 24 |
| 5 | CONCLUSÕES E CONSIDERAÇÕES FINAIS | 27 |
| | REFERÊNCIAS | 28 |

1 Introdução

A utilização de métodos computacionais para a resolução de problemas de otimização combinatória tem contribuído consideravelmente para o aprimoramento de processos produtivos. Seu emprego é essencial na atividade econômica, promovendo melhores resultados para alocação de recursos escassos. O problema de roteamento, por exemplo, que encontramos no rol das atividades da logística, pode ser modelado a partir do problema do caixeiro viajante (PCV), cujo objetivo é encontrar a menor distância a ser percorrida em um conjunto de n cidades, sendo cada cidade visitada apenas uma vez, Golbarg e Luna (2000). A modelagem desse e de outros problemas combinatórios seguem conforme Equação 1.1, de acordo com Ribeiro e Karas (2013):

$$\begin{aligned} & \text{otimizar } f(x) \\ & \text{sujeito a } x \in \Omega \end{aligned} \tag{1.1}$$

onde $f : \mathbb{R}^n \rightarrow \mathbb{R}$ é uma função arbitrária e $\Omega \subset \mathbb{R}^n$ é um conjunto qualquer.

Problemas desta natureza são classificados como NP-difíceis, ou seja, são problemas que não possuem algoritmos capazes de encontrar a solução ótima em tempo polinomial. Para um grande número de variáveis, a verificação das soluções é inconcebível ainda para um supercomputador. No entanto, apesar da inviabilidade de se obter a solução ótima, desenvolveu-se nos últimos anos técnicas capazes de gerar soluções aproximadas em tempo de execução aceitável. Essas técnicas, denominadas heurísticas e meta-heurísticas, podem ser compreendidas como algoritmos construídos a partir de processos intuitivos que reúnem conceitos das áreas de otimização e inteligência artificial Marcone (2011).

Nesse sentido, a otimização de processos logísticos, através do emprego de meta-heurísticas para resolução de problemas, tem possibilitado a maximização da rentabilidade da força de vendas através da minimização dos deslocamentos de veículos Virgillito (2017). Dentre as técnicas utilizadas para a resolução do PCV de maneira aproximada, podemos destacar as meta-heurísticas *Simulated Annealing* (SA) e Algoritmo Genético (AG), que serão objeto do presente estudo.

1.1 Objetivos

O objetivo geral deste trabalho é avaliar a performance das metas-heurísticas SA e AG na resolução de problemas de roteamento com elevadas dimensões que, a princípio,

não podem ser resolvidos por métodos combinatórios capazes de fornecer a solução ótima em um tempo razoável.

1.1.1 Objetivos Específicos

- Analisar o desempenho da técnica de SA em problemas de roteamento de grande porte;
- Analisar o desempenho da técnica de AG em problemas de roteamento de grande porte;
- Comparar a execução e as soluções geradas pelos métodos SA e AG;

1.1.2 Justificativa

O presente trabalho surge na necessidade de comparar a performance das meta-heurísticas clássicas SA e AG no PCV de alta complexidade. Trata-se de um problema presente no dia-a-dia das atividades produtivas e administrativas, principalmente no setor logístico, conforme mencionado anteriormente.

Considerando que, na prática, a quantidade de variáveis não é um parâmetro que pode ser controlado, é imprescindível realizar testes exaustivos e comparativos em ambientes de alta complexidade, a fim de avaliar e comparar o desempenho das técnicas e, se possível, indicar qual a mais indicada para o problema selecionado.

Além do enfoque logístico, é importante observar que as meta-heurísticas podem ser facilmente adaptadas a outros problemas de natureza similar como de Sequenciamento de Máquinas, Layout de Fábrica, Mix de Produção, Fluxo de Caixa, etc. Desta forma, os resultados do estudo também podem ser considerados em uma possível decisão que envolva a utilização dos algoritmos na resolução destes problemas.

2 Referencial Teórico

2.1 Logística e o Problema do Caixeiro Viajante

Em um momento de crescimento econômico e de expansão das atividades comerciais, a logística é parte essencial das estratégias corporativas que buscam otimizar a movimentação de mercadorias, insumos e pessoas. Com este objetivo, ela pode fornecer inúmeras ferramentas para entregar aos clientes níveis de serviço por eles demandados no devido momento, custo, lugar e condições Faria e Costa (2005).

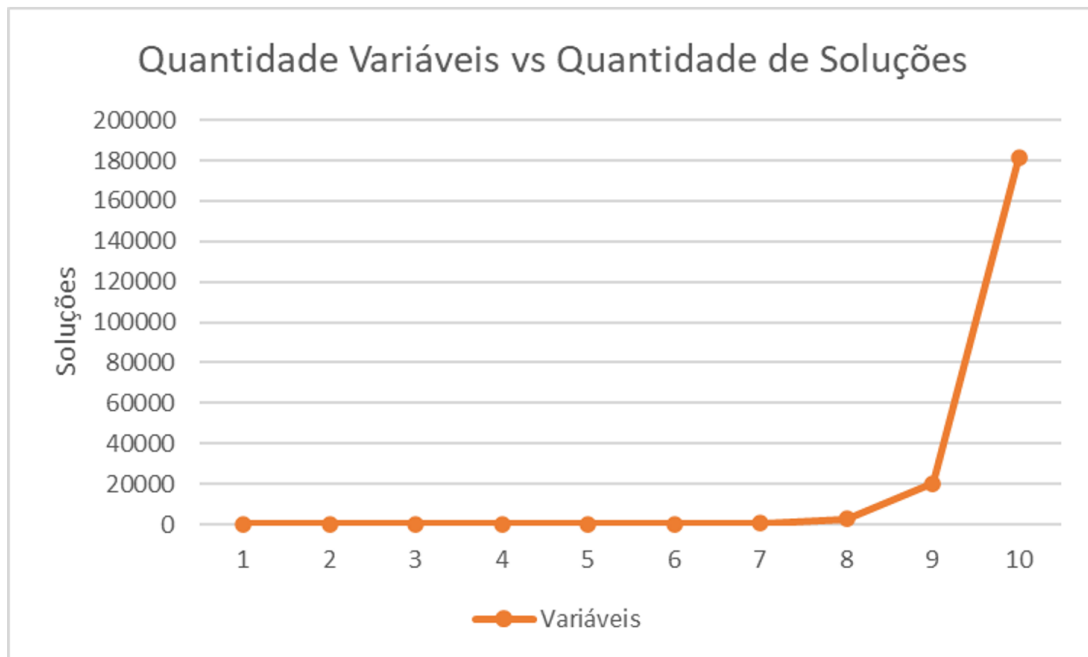
De acordo com Londe B. J. (1989), pode-se definir logística como o processo de planejar, implementar e controlar de maneira eficiente o fluxo e a armazenagem de produtos, bem como os serviços e informações associadas, cobrindo desde o ponto de origem até o ponto de consumo com a finalidade de atender aos requisitos do consumidor. Além disso, de acordo com Novaes (2016), a logística busca incorporar:

- Prazos previamente acertados e cumpridos integralmente, ao longo de toda a cadeia de suprimento;
- Integração efetiva e sistêmica entre todos os setores da empresa;
- Integração efetiva e estreita (parcerias) com fornecedores e clientes;
- Busca da otimização global, envolvendo a racionalização dos processos e a redução de custos em toda a cadeia de suprimentos;
- Satisfação plena do cliente, mantendo o nível de serviço preestabelecido e adequado.

A busca pela otimização global, envolvendo a racionalização dos processos e a redução de custos em toda a cadeia de suprimentos, pode ser obtida a partir do emprego de técnicas computacionais. O Problema de Roteamento de Veículos (PRV), por exemplo, pode ser modelado a partir do PCV, onde busca-se determinar a menor rota possível, dentre um conjunto de restrições Ballou (2009). Outras configurações podem ser incorporados ao modelo considerando aspectos próprios da atividade logística como recursos disponíveis, limites de tempo impostos pela jornada de trabalho, restrições de trânsito, etc.

A solução ótima ou determinística para o PCV depende, principalmente, do número de soluções possíveis, dado por $(n - 1)!/2$, onde n representa o número de cidades a serem visitadas. O gráfico da Figura 1 ilustra a evolução das possibilidades a serem avaliadas de acordo com o aumento de n .

Figura 1 – Evolução do número de soluções de acordo com o aumento de cidades a serem visitadas no PCV.



Fonte: Autor.

Para um grande número de cidades, a enumeração completa de todas as soluções é inconcebível, ainda para um supercomputador atual. Conforme Ballou (2009), o PCV apresenta dificuldades crescentes de resolução, em termos de tempo de processamento, quando o número de localidades aumenta. Problemas de pequeno porte são facilmente resolvidos enquanto problemas maiores requerem tempos de processamento mais elevados. Como não existem algoritmos que resolvam o problema em tempo polinomial, o PCV é classificado como um problema NP-difícil Marcone (2011). Uma alternativa é buscar soluções aproximadas através das heurísticas e meta-heurísticas, conforme descrito a seguir.

2.2 Meta-heurística *Simulated Annealing*

Conforme Marcone (2011), a técnica de SA baseia-se em uma analogia com a termodinâmica ao simular o resfriamento de um conjunto de átomos aquecidos. O controle sistemático da temperatura permite a obtenção de um material com estrutura e propriedades desejadas. A cada redução na agitação das moléculas, a natureza se encarrega de determinar novas configurações atômicas, prevalecendo a de menor consumo energético.

Na meta-heurística SA, a temperatura é controlada por meio de uma função $T(t)$ e, a cada valor, são avaliados N candidatos gerados a partir de um algoritmo de busca local, cujo objetivo é explorar vizinhos obtidos a partir de modificações em uma solução exis-

tente. De acordo com Reeves (2010), o SA pode ser entendido como um aprimoramento desta categoria de algoritmos, de busca local, que tendem a convergir para um mínimo local. O SA, no entanto, com o propósito de explorar o espaço solução com efetividade, evita ótimos locais a partir de uma função de aceitação $A(T)$, dependente de T , que pode aceitar vizinhos cujo valor não seja, necessariamente, melhor do que a solução previamente encontrada Eglese (1990). O sistema é otimizado na medida em que T é reduzido, conduzindo ao estágio de equilíbrio, onde nenhuma mudança tende a ocorrer Kirkpatrick (1983). Desta forma, o algoritmo explora inúmeras áreas do espaço de soluções até convergir para o resultado final.

$A(T)$ e $T(t)$ são responsáveis por ditar a dinâmica do sistema de recozimento. Quanto mais rápido a redução do valor de T , menor o tempo de convergência. O valor retornado por $A(T)$ é confrontado com um segundo valor, gerado aleatoriamente, que determina a aceitação ou não do vizinho obtido. Este comportamento pode ser visualizado no pseudo-código abaixo, estruturado para minimização do valor da solução, conforme proposto por Eglese (1990).

```

Selecionar estado inicial  $i \in S$ ;
Selecionar temperatura inicial  $T > 0$ ;
Definir contador de temperatura  $t = 0$ ;
Repetir
    Definir contador de repetição  $n = 0$ ;
    Repetir
        Gerar estado  $j$ , vizinho de  $i$ ;
        Calcular  $\Delta = f(j) - f(i)$ ;
        Se  $\Delta < 0$  então  $i := j$ 
        Senão  $random(0, 1) < exp(-\Delta/T)$  então  $i := j$ ;
         $n := n + 1$ ;
    Até  $n = N(t)$ ;
     $t := t + 1$ ;
     $T := T(t)$ ;
Até critério de parada verdadeiro.

```

As três primeiras linhas definem o estado inicial, ponto de partida para a busca local, a temperatura inicial e o parâmetro que irá controlar o fluxo de execuções. A cada subciclo, é realizado uma busca local modificando-se o elemento do estado atual. O vizinho obtido é então comparado com o estado atual, calculando-se a diferença entre as soluções. Se Δ for negativo, o vizinho encontrado passa a ser o estado atual. Caso positivo, a substituição é realizada de acordo com o a condição de aceitação: $random(0, 1) < exp(-\Delta/T)$ então $i := j$.

2.3 Algoritmo Genético

Assim como o método de SA, o AG tem sua inspiração em processos que ocorrem na natureza. De acordo com Marcone (2011), trata-se de uma meta-heurística que se fundamenta em uma analogia com processos naturais de evolução nos quais, dada uma população, os indivíduos com características genéticas melhores têm maiores chances de sobrevivência e de produzir filhos cada vez mais aptos, enquanto indivíduos menos aptos tendem a desaparecer. Ainda, segundo definição de Goldberg (2006), trata-se de uma busca estocástica inspirada na evolução, possibilitando o alcance da solução ótima ou aproximada.

Em uma aplicação geral do AG, o primeiro passo é gerar um conjunto de soluções que vai compor a primeira população. Essa população será avaliada e enquanto o critério de parada não for alcançado, novas gerações são geradas a partir da combinação de elementos com melhores resultados. Além disso, ainda são realizadas mutações, com finalidade de alterar, de maneira aleatória, um elemento ou um conjunto de elementos de uma solução ou conjunto de soluções, de forma a garantir uma variedade genética das populações. Segue abaixo o pseudocódigo para implementação de um AG, conforme Reeves (2010).

Selecionar uma população inicial de cromossomos;

Enquanto condição de parada não for satisfeita

Repetir

Se condição de cruzamento satisfeita **então**

 Selecionar cromossomos pais;

 Definir parâmetros de cruzamento;

 Realizar cruzamento;

Se condição de mutação satisfeita **então**

 Escolha pontos de mutação;

 Realizar mutação;

 Avaliar a qualidade da solução obtida;

Até número total de elementos;

Selecionar nova população;

Finalizar ciclo;

3 Metodologia

3.1 Classificação da Pesquisa

De acordo com Venazi et al. (2016), trata-se de uma pesquisa direta, quantitativa e descritiva. Os algoritmos, objeto deste estudo, foram implementados e os resultados coletados para análise.

3.2 Parâmetros

A fim de viabilizar o estudo, definiu-se uma parametrização baseada em elementos diretamente associados aos conceitos das meta-heurísticas:

- Cruzamento (AG): representa a quantidade de cruzamentos realizados para a obtenção de uma nova geração;
- Mutação (AG): define o grau de modificações aleatórias realizadas nos elementos da próxima geração;
- Alpha (SA): determina a velocidade de queda da temperatura;
- Temperatura Inicial (SA): representa o grau inicial de agitação do sistema, período de aceitação de soluções que não são, necessariamente, melhores.

3.3 Gap

Para avaliar o grau de proximidade entre as soluções do sistema, foi utilizado o indicador de *GAP*, calculado conforme Equação 3.1, onde *RE* corresponde à solução encontrada pela meta-heurística e *RO* à solução ótima.

$$GAP = (RE - RO)/(RE) \quad (3.1)$$

3.4 Matriz de distâncias

Para a execução dos algoritmos em um cenário de grande porte foram geradas, aleatoriamente, duas matrizes responsáveis por fornecer a distância entre as cidades. As matrizes foram utilizadas para determinar a distância entre uma cidade e a outra, com o valor de linha representando a origem e o valor de coluna representando o destino. Um exemplo é apresentado na Figura 2.

Figura 2 – Exemplo matriz de distâncias.

| | A | B | C | D |
|---|---|---|---|---|
| A | 0 | 2 | 1 | 9 |
| B | 2 | 0 | 3 | 4 |
| C | 1 | 3 | 0 | 7 |
| D | 9 | 4 | 7 | 0 |

Fonte: Autor.

Para o exemplo da Figura 2, a distância entre a cidade A e a cidade D é igual a 9. Como o caminho de D para A é considerado o mesmo, a matriz é dita simétrica.

3.5 Implementação das meta-heurísticas

Os algoritmos foram implementados através da linguagem de programação C++, utilizando a plataforma de desenvolvimento DevC++. Os programas gerados foram executados em um computador Intel(R) Core(TM) i7-1165G7 @2.80GHz 1.69 GHz com memória RAM de 16 gigabytes e sistema operacional Windows 10. Ambos utilizaram a mesma estrutura de dados responsável por manipular as rotas, realizando operações de cálculo de distância e permutação. Demais estruturas específicas dos algoritmos foram implementadas separadamente.

3.6 *Simulated Annealing*

A implementação do SA seguiu conforme pseudo-código e demais recomendações especificadas em Eglese (1990) e Kirkpatrick (1983).

3.6.1 Busca Local

O método de busca local implementado baseia-se na permutação de 15% dos elementos da estrutura de dados responsável por armazenar o percurso. Em uma instância de 200 cidades, por exemplo, a troca é realizada, de maneira aleatória, em 30 componentes desse vetor. A cada nova execução, novas rotas são geradas.

3.6.2 Função de temperatura

O valor da temperatura é atualizado a cada ciclo de execução conforme Equação 3.2:

$$T = \alpha T \quad (3.2)$$

com valor de α conforme a parametrização.

3.6.3 Função de aceitação

O valor de retorno da função de aceitação é comparado com uma função de distribuição normal que gera números entre (0, 1). Sua implementação segue conforme Equação 3.3:

$$A = e^{-\Delta/T} \quad (3.3)$$

onde Δ corresponde a diferença de distância e T a temperatura do ciclo.

3.7 Algoritmo Genético

A implementação da meta-heurística AG seguiu conforme o pseudo-código e demais recomendações especificadas em Reeves (2010).

3.7.1 Operação de cruzamento

A função responsável pelo processo de cruzamento recebe duas rotas como parâmetros e combina seus elementos em uma única rota. Valores gerados aleatoriamente determinam a parcela da primeira rota que irá permanecer inalterada no resultado final e o restante dos elementos são distribuídos conforme a segunda rota. Essa manipulação pode ser facilmente compreendida a partir da Figura 3:

Figura 3 – Operação de cruzamento de rotas.



Fonte: Autor.

3.7.2 Operação de mutação

O procedimento de mutação foi implementado através da permutação de um percentual fixo dos elementos que compõem a rota, cerca de 5%. A parametrização, por outro lado, define a probabilidade dos elementos gerados sofrerem mutação.

3.7.3 Gerador de populações

A quantidade de rotas para cada população foi fixado em 300 elementos. A primeira população é gerada, de forma aleatória, a partir de operações básicas de permutação e, a cada ciclo, são selecionados os melhores elementos que podem seguir para a próxima geração ou submetidos a operação de cruzamento.

4 Apresentação e Discussão dos Resultados

4.1 Validação da Implementação

Com o propósito de validar a implementação dos algoritmos, foi utilizado como base um problema cuja solução ótima já é conhecida. O problema selecionado foi abordado em Magalhães e Gomes (2016) onde é determinado a menor rota para a distribuição de geleias de mocotó nos estabelecimentos da região do Vale do Aço, que compreende um conjunto de 35 cidades. A matriz de distâncias utilizada pelos autores está disponível e foi empregada no processo de validação. A solução ótima encontrada pelos autores foi obtida utilizando o *software* CPLEX 12.6.2 e equivale à uma distância de 36,74 *km*. O valor mais próximo encontrado pelo SA foi 39,66 *km* e pelo AG foi de 38,26 *km*, resultando em *GAPs* iguais a 7,3% e 3,9%, respectivamente. O resultado das parametrizações realizadas, estão disponibilizados nas Tabelas 1 e 2

| Alpha (%) | Temperatura | Distância (Km) | Tempo (s) |
|-----------|-------------|----------------|-----------|
| 97 | 0,5 | 42,34 | 17,36 |
| 97 | 1 | 48,05 | 17,93 |
| 97 | 10 | 50,16 | 16,82 |
| 98 | 0,5 | 45,93 | 17,91 |
| 98 | 1 | 54,09 | 18,01 |
| 98 | 10 | 55,41 | 18,03 |
| 99 | 0,5 | 53,75 | 18,08 |
| 99 | 1 | 39,66 | 18,14 |
| 99 | 10 | 52,92 | 18,36 |

Tabela 1 – Resultados parametrização *Simulated Annealing*.

| Cruzamento (%) | Mutação (%) | Distância (Km) | Tempo (s) |
|----------------|-------------|----------------|-----------|
| 20 | 1 | 40,04 | 65,11 |
| 20 | 2 | 46,80 | 65,28 |
| 20 | 3 | 47,02 | 57,62 |
| 22 | 1 | 38,26 | 59,45 |
| 22 | 2 | 43,00 | 57,83 |
| 22 | 3 | 45,32 | 58,15 |
| 24 | 1 | 43,11 | 58,34 |
| 24 | 2 | 43,28 | 59,54 |
| 24 | 3 | 43,13 | 63,00 |

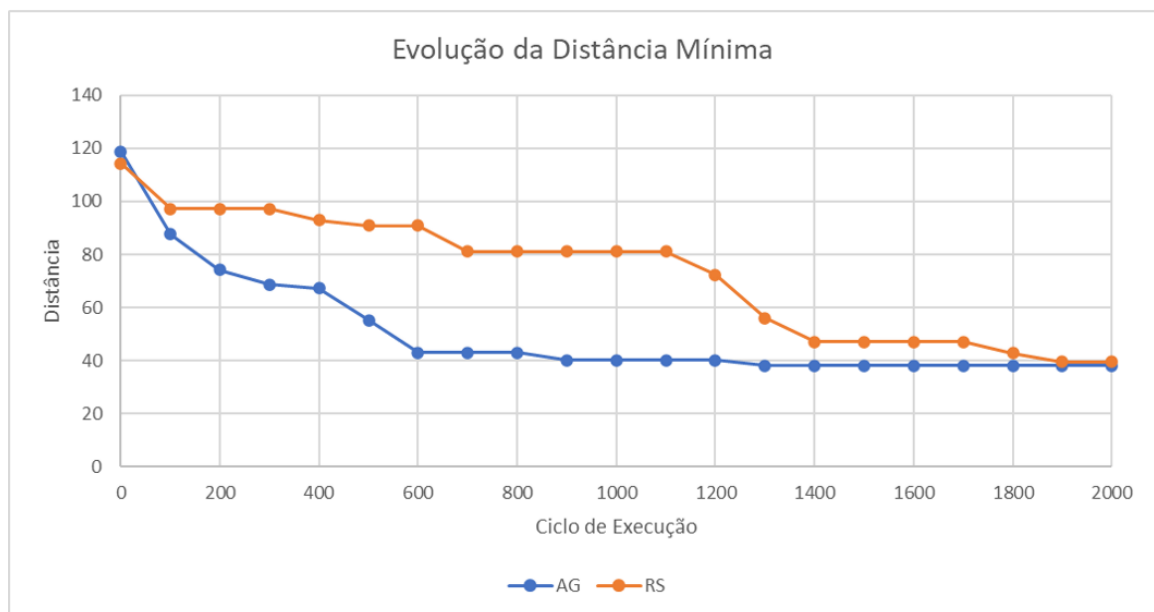
Tabela 2 – Resultado parametrização Algoritmo Genético.

Considerando as taxas de cruzamento selecionadas, o aumento percentual da mutação partindo de 1% resultou, de forma geral, em uma piora do resultado obtido. Já em

relação ao tempo de execução, não se demonstrou sensível as alterações tanto na taxa de cruzamento quanto na taxa de mutação. Já para as parametrizações do SA, conforme a Tabela 1, o aumento de α , partindo de 97%, não resulta na melhora do resultado. Em relação ao valor de temperatura inicial, demonstrou seguir certo padrão para valores de α correspondentes à 97% e 98%, onde o aumento resultou na piora das soluções obtidas. Para α igual à 99% não foi possível identificar tendência dado o resultado obtido com o valor de temperatura igual à 1, que é consideravelmente melhor do que o restante. O tempo total de execução do SA correspondeu à 30% do tempo total gasto com o AG para o problema tratado em Magalhães e Gomes (2016).

Na Figura 4 pode-se visualizar a evolução da distância mínima por ciclo de execução para cada um dos métodos. O AG apresenta uma convergência mais rápida se comparada ao SA. Este comportamento se deve a busca pelo aperfeiçoamento contínuo entre as gerações, onde cada ciclo de execução tende a ser melhor do que o precedeu. Já no SA, a curva de convergência iniciou a consolidação somente após 1.200 ciclos de execução, de maneira abrupta, o que é reflexo direto da estratégia de explorar com efetividade o espaço de soluções antes de se aprofundar em um resultado que talvez corresponda a um mínimo local.

Figura 4 – Evolução de execução.



Fonte: Autor.

4.2 Cenário 01: 100 Localidades

Conforme mencionado anteriormente, os algoritmos implementados foram aplicados em dois cenários de grande porte, com 100 e 200 localidades, calculados a partir de uma

matriz de distância gerada aleatoriamente. Os resultados obtidos pelos métodos para o cenário com 100 localidades são apresentados nas Tabelas 3 e 4.

| Cruzamento (%) | Mutação (%) | Distância (Km) | Tempo (s) |
|-----------------------|--------------------|-----------------------|------------------|
| 20 | 1 | 2835 | 326,11 |
| 20 | 2 | 2759 | 300,56 |
| 20 | 3 | 2968 | 318,13 |
| 22 | 1 | 2984 | 317,67 |
| 22 | 2 | 3022 | 382,71 |
| 22 | 3 | 3070 | 317,76 |
| 24 | 1 | 3262 | 311,06 |
| 24 | 2 | 3450 | 301,79 |
| 24 | 3 | 3206 | 301,37 |

Tabela 3 – Resultado execução 100 localidades - AG.

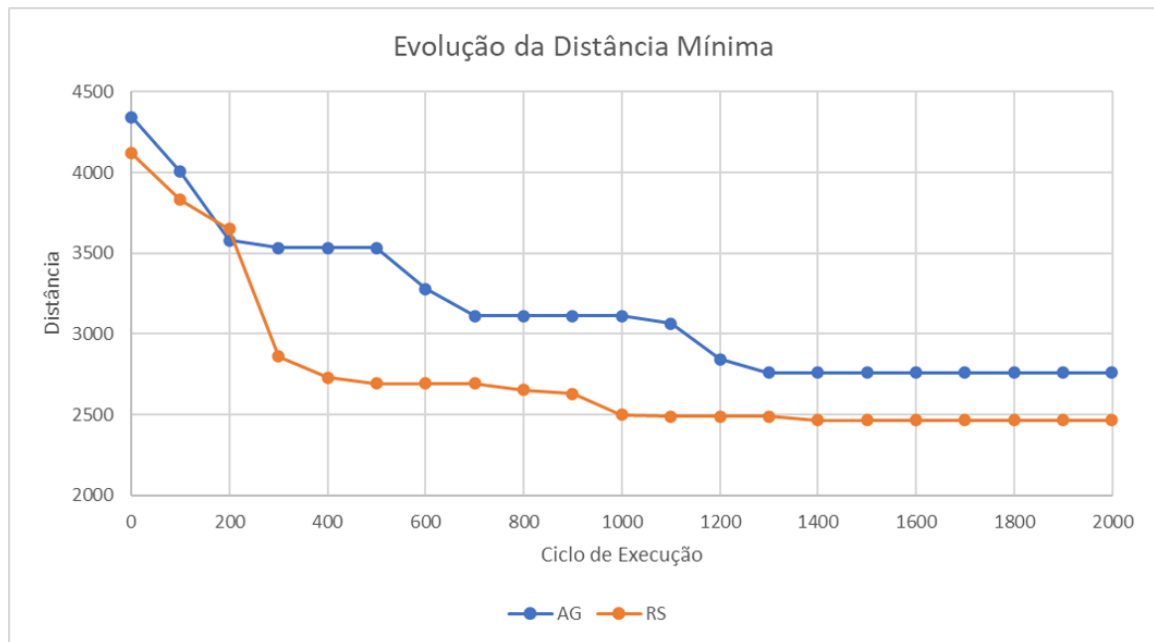
| Alpha (%) | Temperatura | Distância (Km) | Tempo (s) |
|------------------|--------------------|-----------------------|------------------|
| 99 | 1 | 2605 | 44,46 |
| 99 | 0,5 | 2482 | 44,23 |
| 99 | 10 | 2659 | 45,18 |
| 98 | 1 | 2621 | 43,91 |
| 98 | 0,5 | 2603 | 43,70 |
| 98 | 10 | 2572 | 43,86 |
| 97 | 1 | 2517 | 43,65 |
| 97 | 0,5 | 2465 | 43,39 |
| 97 | 10 | 2620 | 43,66 |

Tabela 4 – Resultado execução 100 localidades - SA.

Para a instância com 100 localidades, pode-se observar que o aumento da taxa de cruzamento e do percentual de mutação do AG resultou em uma piora no resultado obtido. O tempo de execução, seguindo o comportamento identificado anteriormente, não foi impactado com as parametrizações. Já nos resultados obtidos pelo SA, não é possível estabelecer um padrão para a parametrização, sem variação considerável no seu tempo de execução. No gráfico da Figura 5, pode-se visualizar a convergência gradual para o AG enquanto para o SA nota-se um comportamento de redução abrupta em pequenos intervalos de execução.

Não é possível afirmar a distância que esses resultados estão da solução ótima mas, para efeitos comparativos, considerando a melhor solução obtida como a solução ótima, temos que para essa instância de 100 variáveis, o GAP entre as soluções foi de 10%.

Figura 5 – Evolução de execução para 100 localidades.



Fonte: Autor.

4.3 Cenário 02: 200 Variáveis

Os resultados para o cenário com 200 localidades seguiram o mesmo comportamento do cenário com 100 localidades. O SA apresentou melhores soluções que o AG. Desta vez, no entanto, não foi possível estabelecer um padrão de acordo com a parametrização. Os resultados obtidos pelos métodos para o cenário com 200 localidades são apresentados nas Tabelas 5 e 6.

| Cruzamento (%) | Mutação (%) | Distância (Km) | Tempo (s) |
|----------------|-------------|----------------|-----------|
| 20 | 1 | 7760 | 1745,84 |
| 20 | 2 | 7639 | 1760,44 |
| 20 | 3 | 7998 | 1779,65 |
| 22 | 1 | 7700 | 1294,70 |
| 22 | 2 | 7779 | 1285,98 |
| 22 | 3 | 7919 | 1762,21 |
| 24 | 1 | 7835 | 1723,69 |
| 24 | 2 | 8028 | 1423,22 |
| 24 | 3 | 7827 | 1213,02 |

Tabela 5 – Resultado execução 200 localidades - AG.

| Alpha (%) | Temperatura | Distância (Km) | Tempo (s) |
|-----------|-------------|----------------|-----------|
| 99 | 1 | 6572 | 119,95 |
| 99 | 0,5 | 5938 | 104,60 |
| 99 | 10 | 6403 | 107,37 |
| 98 | 1 | 6261 | 102,59 |
| 98 | 0,5 | 6442 | 102,97 |
| 98 | 10 | 6080 | 103,34 |
| 97 | 1 | 6027 | 102,37 |
| 97 | 0,5 | 6124 | 101,28 |
| 97 | 10 | 6044 | 102,23 |

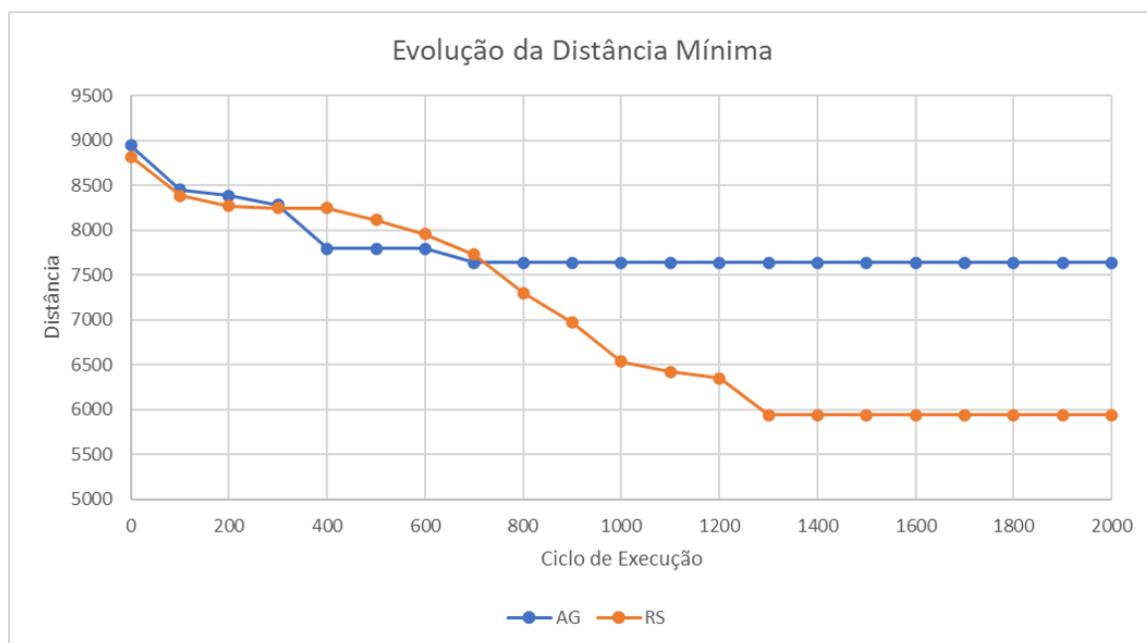
Tabela 6 – Resultado execução 200 localidades - SA.

No cenário com 200 variáveis, o tempo de execução torna-se fator de atenção. O Algoritmo Genético apresentou um tempo médio de execução de 1.554 segundos, quase 15 vezes o tempo médio de execução do *Simulated Annealing* de 105 segundos. O aumento pode ser explicado pela quantidade de manipulações realizadas nos vetores de cruzamento.

A menor distância obtida no SA foi de 5.938 Km, que contrasta com o AG, com menor distância de 7.639 Km. Seguindo o raciocínio das últimas análises, o GAP entre as duas soluções foi de 22%. O aumento do GAP nos cenários pode ser explicado pelo parâmetro que especifica o tamanho da população, que permaneceu fixo e limita o grau de “variedade genética” entre as gerações.

Na Figura 6, observa-se a evolução das melhores instâncias de execução. Ambos algoritmos minimizam a solução até 800 ciclos de execução. Após este momento, o AG permanece “preso” à um mínimo local enquanto o SA encontra melhores valores até 1.300 ciclos de execução.

Figura 6 – Evolução de execução para 200 localidades.



Fonte: Autor.

5 Conclusões e considerações finais

Para a implementação e parâmetros selecionados, o SA demonstrou-se melhor pelo fato de, além de possuir uma aplicação prática com poucas linhas de código, foi capaz de fornecer resultados melhores em um tempo reduzido de processamento. Em relação ao AG, a manipulação constante de vetores no processo de cruzamento demandou maior tempo de execução, que cresceu com o aumento de localidades.

Para o PCV, considerando a implementação, parametrização e resultados do presente estudo, o SA é mais indicado do que o AG. Vale ressaltar que existe uma infinidade de resultados que podem ser obtidos através de diferentes ajustes e parâmetros, não sendo possível determinar de forma definitiva, para diferentes conjuntos de dados, se uma meta-heurística sempre será melhor do que a outra.

Os parâmetros, fixados na implementação a fim de simplificação da análise, podem ser objeto de estudo em análises futuras utilizando ferramentas de calibragem, como o *iRace* da linguagem R. Abaixo lista-se pontos para serem abordados oportunamente:

- Quantidade de elementos permutados no processo de mutação;
- Tamanho da população;
- Quantidade de elementos permutados no processo de busca local;
- Número de iterações para cada temperatura.

Além disso, outras estratégias de implementação podem ser consideradas nos algoritmos:

- Troca do algoritmo de Busca Local;
- Método de seleção dos melhores elementos da população;
- Função de Temperatura;
- Função de Aceitação.

Referências

- BALLOU, R. H. *Gerenciamento da Cadeia de Suprimentos-: Logística Empresarial*. [S.l.]: Bookman Editora, 2009.
- EGLESE, R. W. Simulated annealing: a tool for operational research. *European journal of operational research*, Elsevier, v. 46, n. 3, p. 271–281, 1990.
- FARIA, A. C. d.; COSTA, M. d. F. G. d. *Gestão de custos logísticos*. São Paulo: Atlas, 2005.
- GOLBARG, M. C.; LUNA, H. P. L. *Otimização combinatória e programação linear*. Editora CAMPUS, Rio de Janeiro, 2000.
- GOLDBERG, D. E. *Genetic algorithms*. Pearson Education India, 2006.
- KIRKPATRICK. Optimization by simulated annealing. *science*, American association for the advancement of science, v. 220, n. 4598, p. 671–680, 1983.
- LONDE B. J., C. M. C. L. Partnerships in providing customer service: a third-party perspective. oak brook: The council of logistics managements. 1989.
- MAGALHÃES, D. C.; GOMES, G. P. *Logística de distribuição em uma empresa de geléia de mocoto na região do vale do aço*. 2016.
- MARCONE, J. F. S. *Inteligência computacional para otimização*. 2011.
- NOVAES, A. *Logística e gerenciamento da cadeia de distribuição*. [S.l.]: Elsevier Brasil, 2016.
- REEVES, C. R. Genetic algorithms. In: *Handbook of metaheuristics*. [S.l.]: Springer, 2010. p. 109–139.
- RIBEIRO, A. A.; KARAS, E. W. *Otimização contínua: Aspectos teóricos e computacionais*. 2013.
- VIRGILLITO, S. B. *Pesquisa Operacional*. [S.l.]: Saraiva Educação SA, 2017.