

SIMULATED ANNEALING APLICADO AO PROBLEMA DO CAIXEIRO VIAJANTE: UM ESTUDO DE CASO.

Maria Clara de Oliveira Gê¹, Matheus da Silva Menezes², Vanessa Elionara Souza Ferreira³

Resumo: Dentre os problemas de otimização combinatória, o Problema do Caixeiro Viajante possui a complexidade da classe NP-Difícil, esses problemas apresentam uma peculiaridade em relação aos outros, que é o de determinar a solução ótima em um tempo computacional razoável, visto que o espaço de soluções cresce exponencialmente com as dimensões do problema. Nesses casos, a utilização de metaheurísticas surge como uma alternativa para obter soluções candidatas, muito próximas da solução ótima e, em alguns casos, a própria solução ótima global. Este artigo apresenta a implementação da metaheurística *Simulated Annealing* adaptada ao Problema do Caixeiro Viajante contido em uma empresa distribuidora de laticínios, localizada no estado do RN. O objetivo é validar a eficiência do método proposto, considerando o desempenho computacional gasto em cada execução e a qualidade das soluções encontradas em cada iteração. Os resultados obtidos neste trabalho foram postos em comparação com os resultados de estudos prévios através do modelo exato em [1] e de algoritmos genéticos e meméticos em [2] e [3]. Com a análise comparativa, pode-se aferir que a metaheurística proposta constitui-se de um método de obtenção de boas soluções para problemas de otimização difíceis, tanto em relação a qualidade das soluções obtidas quanto ao tempo de execução do algoritmo.

Palavras-chave: Problema do Caixeiro Viajante; metaheurísticas; *Simulated Annealing*.

1. INTRODUÇÃO

Muitos problemas em engenharia, planejamento, fabricação e distribuição, podem ser modelados como o de minimizar ou maximizar uma função. Esta classe dos chamados problemas de otimização, são frequentemente identificados no cotidiano das organizações e envolvem diferentes variáveis interligadas em um único problema, com diversos cenários para análise e tomada de decisão. A dificuldade constitui-se em encontrar dentre todas as soluções, a solução globalmente ótima que atenda todos os critérios envolvidos [4]. A Pesquisa Operacional é uma área de conhecimento interdisciplinar que congrega algoritmos e técnicas utilizadas na resolução de problemas reais em prol de uma tomada de decisão eficaz. Uma forma de resolução que engloba esse campo de estudo é a programação matemática, que permite examinar inúmeras configurações viáveis do problema em questão e oferecer a melhor solução, respeitando um conjunto de restrições. Essas técnicas consagraram-se em face à sua grande utilidade na resolução de problemas de otimização combinatória; dentre os quais pode-se citar: Problemas de Fluxo em Redes e Logística de Distribuição; o Problema do Caixeiro Viajante, o Problema da Mochila, o Problema de Transporte, o Problema de Alocação, dentre outros [5].

O Problema do Caixeiro Viajante - PCV consiste em visitar um conjunto finito de cidades, de tal forma que inicie e termine a rota na mesma cidade, passando pelas demais uma única vez ao menor custo possível [5]. O PCV possui larga aplicabilidade no mundo real, é de fácil compreensão e descrição, porém classificado como NP-Difícil [6], pois o espaço de soluções cresce exponencialmente com as dimensões do problema. Instâncias conectadas com n cidades, têm $(n - 1)!$ rotas possíveis e n operações matemáticas são processadas para cada rota. Diante disso, estratégias que utilizam metaheurísticas estão mostrando resultados cada vez mais promissores, sem a necessidade de alocar grandes recursos computacionais para tal. Cada metaheurística detém de uma estratégia de busca, porém o objetivo comum é escapar dos mínimos locais, a fim de avançar a exploração no espaço de busca por soluções ainda melhores [7].

Simulated Annealing, é uma metaheurística baseada na analogia com o processo físico do recozimento de certos materiais (*annealing*) proposto inicialmente por [8], e [9] sugeriu sua utilização no ambiente computacional, o método apresenta como critério de convergência princípios termodinâmicos. Estruturas atômicas são formadas em função do processo de aquecimento, seguido do resfriamento dos materiais. Com uma lenta e gradativa diminuição da temperatura, cada configuração apresenta um nível de energia. O processo termina quando o ponto de solidificação é atingido, apresentando uma configuração de energia mínima. Para a modelagem matemática, temos que: o estado do sistema termodinâmico corresponde à solução atual do problema combinatório; a equação de equilíbrio de energia para o sistema termodinâmico representa a função objetivo; estados do sistema físico são equivalentes a soluções do espaço de busca; a perturbação de um estado físico relaciona-se com a seleção de uma solução vizinha no espaço de buscas; um resfriamento rápido do sistema

físico trata-se de um ótimo local e o estado fundamental de um sistema de partículas é equivalente ao ótimo global de um problema combinatório [9].

Atualmente, um grande percentual de custos para empresas está na distribuição de produtos através da atividade logística; dessarte, as organizações buscam processos para otimizar a distribuição e assim, reduzir custos. O problema abordado nesta pesquisa, caracteriza-se como uma aplicação prática e generalização do Problema do Caixeiro Viajante, que é o Problema de Roteamento de Veículos (PRV). A Associação dos Pequenos Agropecuaristas do Sertão de Angicos (APASA), realiza entregas de produtos diariamente em um conjunto de cidades do Estado do Rio Grande do Norte. Deve-se encontrar uma rota que passe exatamente uma única vez por cada cidade, de tal forma que minimize o tempo total do percurso ou a distância total da rota. Os objetivos deste trabalho são: implementar o algoritmo *Simulated Annealing* adaptado ao problema da APASA; validar a eficiência do método proposto, considerando o desempenho computacional gasto em cada execução e a qualidade das soluções encontradas em cada iteração; analisar os resultados obtidos comparando-os com trabalhos contidos na literatura, os quais resolvem o mesmo PCV através de métodos exatos e aproximativos.

Além desta corrente introdução, os demais tópicos do artigo estão organizados da seguinte forma: a seção 2 é dedicada ao embasamento teórico pertinente ao problema em questão; a seção 3 contém o estudo de caso que apresenta a descrição do problema e a metodologia empregada. Por conseguinte, a seção 4 relata os resultados dos experimentos computacionais e realiza uma análise e comparação com resultados de algoritmos propostos em trabalhos existentes na literatura, adaptados parcialmente ao mesmo PCV, desenvolvidos por [1], [2] e [3]. Por fim, a seção 5, consistirá nas considerações finais da pesquisa.

2. REFERENCIAL TEÓRICO

Nesta seção são apresentados os conceitos teóricos pertinentes aos temas abordados no contexto da problemática em questão, tais como: O Problema do Caixeiro Viajante com a formulação matemática de Dantzig-Fulkerson-Johnson; metaheurísticas e *Simulated Annealing*. Para o entendimento dos tópicos abordados neste item, o leitor deve estar familiarizado com os conceitos de grafos, problemas de fluxos em redes, problemas de roteamento de veículos e problemas de transporte. Para maiores informações, ver [5].

2.1. O Problema do Caixeiro Viajante

De acordo com [5], a maioria dos problemas de roteamento estão relacionados a rotas com pontos de oferta e demanda. Esses pontos podem representar cidades, centros de distribuição, depósitos, pontos de coleta, etc. Um tipo de rota denominada de ciclo hamiltoniano, surgiu do jogo *Around the World*, proposto por William Rowan Hamilton. O jogo era feito sobre um dodecaedro, em que cada vértice estava associado a uma cidade. O desafio era encontrar uma rota através dos vértices do dodecaedro que iniciasse e terminasse em uma mesma cidade, sem repetir nenhum vértice. O Problema do Caixeiro Viajante (PCV) foi o primeiro problema de roteirização a ser estudado e pode ser entendido neste contexto.

Ainda de acordo com este autor, o PCV é um clássico problema de otimização combinatória, amplamente explorado na literatura. O problema consiste em visitar um conjunto finito de cidades, de tal forma que inicie e termine a rota na cidade de origem, visitando as demais apenas uma vez. O objetivo é encontrar um circuito hamiltoniano de menor custo, seja através da menor distancia percorrida ou do menor tempo. Esse problema pode ser descrito como um grafo, $G = (N, A)$, onde $N = \{1, \dots, n\}$ representa um conjunto de cidades (nós) e $A = \{1, \dots, n\}$ é a distância ou o tempo entre pares de cidades conectadas (arcos).

Existem várias formulações para o PCV. Este trabalho trata do PCV simétrico e adota a comumente utilizada formulação de Dantzig-Fulkerson-Johnson (DFJ), descrita a seguir:

$$\text{Minimizar } Z = \sum_{j=1}^n \sum_{i=1}^n x_{ij} c_{ij} \quad (1)$$

Sujeito à:

$$\sum_{i=1}^n x_{ij} = 1 \quad \forall j \in N \quad (2)$$

$$\sum_{j=1}^n x_{ij} = 1 \quad \forall i \in N \quad (3)$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1 \quad \forall S \subset N \quad (4)$$

$$x_{ij} \in [0, 1] \quad \forall i, j \in N \quad (5)$$

Onde,

(1) representa a função-objetivo do problema, a qual define a minimização do custo total do caminho, considerando que C_{ij} é o custo de se transpor o arco $(i, j) \in A$ e x_{ij} item (5);

(2) e (3) são as restrições que garantem que cada cidade é visitada apenas uma vez;

(4) representa a restrição que proíbe a formação de subrotas; e,

(5) É a variável binária que assume valor igual a 1, se o arco $(i, j) \in A$ for escolhido para integrar a solução, e 0 em caso contrário.

Para [5], essa formulação mostra o PCV com características de programação linear inteira e destaca sua natureza combinatória. Devido à essa última característica, o PCV é considerado um problema pertencente a classe NP-Difícil, problemas que não possuem solução em tempo polinomial por um algoritmo não-determinístico. Simplificando a abordagem descrita por [6], em problemas da classe NP-Difícil o espaço de busca cresce exponencialmente com as dimensões do problema, portanto problemas de grande porte tornam-se inviáveis se forem utilizados métodos exatos, em virtude do esforço computacional que seria exigido para resolvê-los. Diante desse contexto, estudos nas áreas de Otimização e Inteligência Artificial, viabilizaram a construção das denominadas melhores estratégias, métodos conhecidos como metaheurísticas. Esses algoritmos vêm apresentando soluções parciais e ótimas para problemas de otimização combinatória, como o próprio PCV, e geralmente com um tempo de processamento menor [7].

2.2. Metaheurísticas

De acordo com [11], metaheurística é um tipo de método estruturado, que coleta informações com o objetivo de orquestrar a iteração entre procedimentos de busca local e estratégias para criar um processo capaz de escapar de ótimos locais e realizar uma busca consistente em uma região de soluções viáveis pela solução ótima global. Em [10], define-se a grande importância na aplicabilidade de uma metaheurística, que é o balanço dinâmico entre diversificação ou exploração diversificada, busca em largura ou simplesmente exploração e; intensificação ou exploração focada, busca em profundidade. Um dos desafios na aplicação de uma metaheurística é encontrar o equilíbrio ideal entre diversificação e intensificação.

As metaheurísticas diferenciam-se entre si basicamente pelo mecanismo utilizado para escapar de ótimos locais. Dividem-se em duas categorias, de acordo com o princípio usado para explorar o espaço de soluções: busca local e busca populacional. Os métodos baseados em busca populacional, por sua vez, consistem em manter um conjunto de boas soluções e combiná-las de forma a tentar produzir soluções ainda melhores. Exemplos clássicos de procedimentos desta categoria são os Algoritmos Genéticos, os Algoritmos Meméticos e o Algoritmo Colônia de Formigas. As metaheurísticas baseadas em busca local, a exploração do espaço de soluções é feita por meio de movimentos, os quais são aplicados a cada passo sobre a solução corrente, gerando outra solução promissora em sua vizinhança. Busca Tabu, *Simulated Annealing*, GRASP (*Greedy Adaptive Search Procedures*), Busca em Vizinhança Variável (*Variable Neighborhood Search*) e *Iterated Local Search* são exemplos de métodos que se enquadram nesta categoria [12].

2.3. Simulated Annealing

O método proposto inicialmente por [8] simula o processo de rearranjo atômico de alguns materiais, utilizado na metalurgia, para obtenção de estados de baixa energia de um sólido. O termo *annealing* (ou recozimento) é dado ao processo de aquecimento de um sólido até seu ponto de fusão, seguido de um resfriamento gradual e controlado, de modo que ele atinja um estado de baixa energia. Nesse processo, o resfriamento lento é essencial para a manutenção do equilíbrio térmico no qual os átomos possam reorganizar-se em uma estrutura uniforme de mínima energia. Caso o sólido seja resfriado de forma abrupta, seus átomos poderão formar uma estrutura cristalina frágil e irregular, com alta energia em consequência do esforço interno gasto [13].

O processo físico de recozimento pode ser modelado com sucesso por métodos de simulação. O algoritmo introduzido por [8] é baseado em técnicas de Monte Carlo e gera uma seqüência de estados do sólido como descrito a seguir: dado um estado corrente i do sólido com energia E_i , então o estado subsequente j é gerado aplicando um mecanismo de perturbação que transforma o estado corrente no próximo estado por uma pequena distorção. A energia do próximo estado é E_j . Se a diferença de energia $E_j - E_i$, é menor ou igual a 0, o estado j é aceito como estado corrente. Se a diferença de energia é maior que 0, o estado j é aceito com uma certa probabilidade que é dada por:

$$\exp\left(\frac{E_j - E_i}{k_B T}\right) \quad (6)$$

onde T denota a temperatura do banho quente e k_B uma constante física conhecida como constante de Boltzmann. A regra de aceite descrita acima é conhecida como critério de Metropolis e o algoritmo que a usa é conhecido como algoritmo de Metropolis. Se a redução da temperatura é feita suficientemente lenta, o sólido pode encontrar o equilíbrio térmico em cada temperatura. No algoritmo de Metropolis isto é obtido gerando um grande número de transições em uma determinada temperatura.

O algoritmo *Simulated Annealing*, sugerido em ambiente computacional por [9] e [14], determina uma correlação entre o comportamento dos problemas de otimização combinatória e o processo físico do recozimento de sólidos. Para a modelagem matemática, temos a seguinte analogia: o estado do sistema termodinâmico corresponde à solução atual do problema combinatório; a equação de equilíbrio de energia para o sistema termodinâmico representa a função objetivo; estados do sistema físico são equivalentes a soluções do espaço de busca; a perturbação de um estado físico relaciona-se com a seleção de uma solução vizinha no espaço de buscas; um resfriamento rápido do sistema físico trata-se de um ótimo local e o estado fundamental de um sistema de partículas é equivalente ao ótimo global de um problema combinatório.

Conforme [9], esse método tenta evitar a convergência para um mínimo local, aceitando, às vezes, uma nova solução gerada, mesmo que essa incremente o valor da função objetivo. O aceite ou a rejeição de uma nova solução, é determinado por um critério probabilístico, através de uma função p conhecida por função de aceite. Normalmente, essa função é expressa por:

$$p = \exp \left(\frac{F_{s'} - F_s}{T} \right) \quad (7)$$

Caso $\Delta = F_{s'} - F_s$, for menor que zero, a solução s' será aceita como a nova solução corrente. Caso contrário, a nova solução somente será aceita se:

$$p > \text{random} [0, 1] \quad (8)$$

Para a semelhança com o método original de simulação de recozimento na termodinâmica, têm-se o Δ , correspondente à variação da energia de um estado para outro (dE) e o parâmetro de controle T , que corresponde à temperatura.

O SA procura explorar novas áreas no espaço de soluções do problema em um curto período de tempo e com pouco esforço computacional. Para isso, o algoritmo recebe como entrada parâmetros que permitem ajustá-lo de forma que a execução seja mais efetiva. Para [9], o conjunto de parâmetros a serem determinados compõe o que se chama programa de resfriamento (*cooling schedule*). Um programa de resfriamento especifica uma sequência finita de valores dos parâmetros de controle, isto é:

- a) uma temperatura inicial;
- b) uma função de redução da temperatura;
- c) um critério de parada que especifica a temperatura final;
- d) um número finito de transições em cada temperatura.

Esse programa foi proposto por [9] e permanece largamente utilizado. O ajuste de parâmetros é o ponto mais sensível para o desempenho do algoritmo, tanto em tempo de execução quanto à qualidade da solução. Uma vez encontrados bons parâmetros para determinado problema, poucos ajustes são necessários para outras instâncias do mesmo.

3. ESTUDO DE CASO

Será mostrado a seguir como a metaheurística *Simulated Annealing* foi adaptada às características do PCV contido na empresa distribuidora de laticínios situada na cidade de Angicos/RN.

3.1. Descrição do Problema

A Associação dos Pequenos Agropecuaristas do Sertão de Angicos (APASA), realiza entregas de produtos diariamente em um conjunto de cidades do Estado do Rio Grande do Norte. O problema consiste em determinar a melhor rota para distribuição dos produtos sobre um conjunto de n cidades, a fim de minimizar os custos, seja ao utilizar a rota de menor distância ou a rota que totalize o menor tempo de viagem. Em sua pesquisa, [1] constata que a empresa atua em doze cidades, incluindo o centro de distribuição e os centros consumidores. Para fins de simulação, o número de cidades foi ampliado em [3], e serão abastecidas pelo mesmo centro de distribuição. O Quadro 1 identifica o conjunto de cidades envolvidas no problema, totalizando 48 cidades.

Quadro 1. Identificação das cidades. (Autoria própria)

Nº	Cidade	Nº	Cidade	Nº	Cidade
1	Angicos	17	São Gonçalo do Amarante	33	Monte Alegre
2	São Rafael	18	Pau dos Ferros	34	Nízia Floresta
3	Itajá	19	Pamamirim	35	Nova Cruz
4	Assu	20	Macaíba	36	Touros
5	Jucurutu	21	Macau	37	Tangará
6	Mossoró	22	João Câmara	38	São Paulo do Potengi
7	Natal	23	Ipanguaçu	39	Santo Antônio
8	Afonso Bezerra	24	Currais Novos	40	Taipu
9	Pedro Avelino	25	Ceará-Mirim	41	Upanema
10	Fernando Pedrosa	26	Canguaretama	42	Pedro Velho
11	Santana do Matos	27	Baraúna	43	Parelhas
12	Lajes	28	Apodi	44	Caraúbas
13	Santa Cruz	29	Acari	45	Alexandria
14	Caicó	30	Areia Branca	46	Jardim de Piranhas
15	São Miguel	31	Extremoz	47	São José do Campestre
16	São José do Mipibu	32	Goianinha	48	Poço Branco

Conforme [3], as características do problema remetem a utilização da abordagem do PCV, utilizando apenas um transporte em cada situação, de forma que supra toda demanda. Para o problema em questão, podemos considerar seis simulações distintas: uma única rota, com 48 cidades, e cinco sub-rotas com: 36 cidades, 24 cidades, 12 cidades, 7 e 6 cidades, respectivamente. Como variáveis de minimização da função objetivo, consideramos o custo da distância percorrida (Km) e o custo do tempo gasto (min), com dados fornecidos pelo *Google Maps*. A Figura 1 ilustra a complexidade do problema, localizando no mapa todas as cidades. A cidade de origem e destino é Angicos, identificada com o símbolo de localização e letra “A”, as demais, estão com o ponto de localização.



Figura 1. Mapa mostrando a localização do problema. [3]

O Quadro 2, a seguir, identifica os problemas (instâncias), a unidade de medida a qual o problema trata para encontrar a melhor rota, o número de cidades (n) e as cidades contidas em cada problema.

Quadro 2 – Instâncias (Autoria própria)

Instância	Medida	n	Cidades pertencentes a instância
Problema 1	Km	48	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47 e 48
Problema 2	Min		
Problema 3	Km	36	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36
Problema 4	Min		
Problema 5	Km	24	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24
Problema 6	Min		
Problema 7	Km	12	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12
Problema 8	Min		
Problema 9	Km	7	1, 2, 3, 4, 5, 6, 7
Problema 10	Min		
Problema 11	Km	6	1, 2, 3, 4, 5, 6
Problema 12	Min		

Trabalhos correlatos abordam o mesmo PCV, [2] propôs o uso de um algoritmo genético tradicional (AGT) para encontrar soluções que se aproximem dos resultados ótimos obtidos em [1] via método exato, e [3] apresentou um algoritmo evolucionário híbrido, o algoritmo memético (AM) que utiliza o AG na forma tradicional, com exceção da taxa dos operadores genéticos, que inclui a mutação em sua operação e recebe um único valor para a execução do crossover e mutação e, a execução do operador de busca local. Nesse contexto, este artigo apresenta uma metaheurística aproximativa, denominada *Simulated Annealing*, para verificar a aplicabilidade do algoritmo e validar sua eficiência mediante estudos prévios citados.

3.2. Experimento computacional

O algoritmo proposto neste trabalho foi executado para todas as instâncias apresentadas em 3.1 e, previamente modelado utilizando a plataforma Scilab. Seus parâmetros foram ajustados de acordo com [9], os valores dos parâmetros são apresentados na Tabela 1. Para melhor compreensão, os parâmetros se dividem em: temperatura inicial (T_i), a temperatura mínima a ser utilizada como critério de parada do algoritmo (T_f), a taxa de redução de temperatura (α) e número de passos a cada temperatura (npT).

Tabela 1: Parâmetros utilizados pelo SA. (Autoria própria)

Parâmetro	Valor
T_i	100
T_f	0,01
α	0,98
npT	5

A quantidade de soluções geradas em cada experimento foi determinada de acordo com trabalhos os correlatos [1, 2 e 3], porém com outras estratégias. Para os problemas de 1 a 12, realizou-se experimentos com uma quantidade de soluções de: 25, 50, 75 e 100. Para os problemas 7 e 8: 2x, 4x e 8x o tamanho do problema e para os problemas 9, 10, 11 e 12: 2x e 4x, o tamanho do problema. Os algoritmos foram executados em um Servidor Dell PowerEdge T130, com Processador Intel® Xeon® Quad Core E3-1220 v6, 8GB de memória RAM e HD de 1TB. O sistema operacional é o Ubuntu 18.04.1 LTS e o ambiente computacional é o Scilab, versão 6.0. Cada problema foi executado 10 vezes, totalizando 390 experimentos computacionais.

3.3. Algoritmo proposto

Um fluxograma para representar o método, é apresentado através da Figura 2 e descrito a seguir.

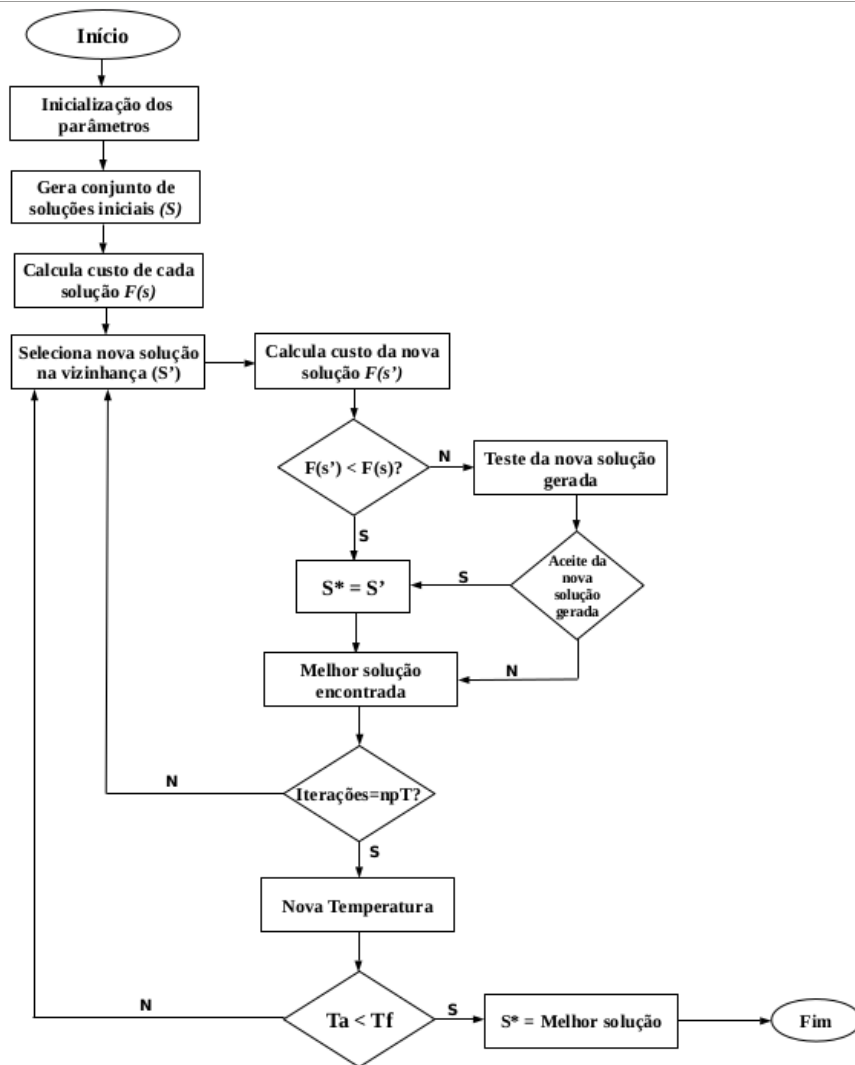


Figura 2. Fluxograma do *Simulated Annealing*. (Autoria própria)

O algoritmo recebe como parâmetros de entrada uma instância do problema, a quantidade de soluções e os parâmetros do programa de resfriamento. Neste trabalho, a solução inicial é gerada aleatoriamente, atendendo as restrições do PCV, ou seja, não permitindo a repetição de uma mesma cidade. A solução inicial é representada por S . Após gerar o conjunto de soluções, o algoritmo calcula o custo de cada solução. Posteriormente, inicia-se um procedimento de Busca Local *Shift* no espaço de soluções. A solução vizinha é representada por S' . A próxima etapa do algoritmo, é verificar a qualidade da nova solução em relação à solução atual, com base na função objetivo. Caso $F(s') < F(s)$, S' é aceita como solução atual, se $F(s') > F(s)$, o fator probabilístico (7) e o teste de aceitação (8), irão determinar se a solução vizinha será ou não aceita como solução atual. Essa escolha por uma eventual pior solução é o mecanismo utilizado pelo SA como tentativa de escapar de ótimos locais. As iterações ocorrem até que uma situação de equilíbrio seja observada. T_i indica a temperatura inicial e T_a , a temperatura atual. Enquanto $T_a > T_f$, um laço de repetição é executado, tendo como mudança de temperatura a taxa decréscimo da temperatura α , e como condição de parada o parâmetro que estabelece a temperatura mínima T_f . O laço de repetição *Iterações* tem como condição de parada o parâmetro npT estabelecido e determina a quantidade de iterações em cada temperatura. Por fim, o algoritmo retorna como saída, a distância ou tempo da melhor rota. É importante destacar que o Algoritmo 1 está voltado à solução de um problema de minimização.

4. RESULTADOS E DISCUSSÕES

Os resultados dos experimentos computacionais são apresentados nesta seção, com a análise individual e comparativa da metaheurística proposta.

A Tabela 2 apresenta os melhores resultados obtidos pelo SA. Para cada problema são designadas cinco colunas apresentando, nesta ordem: a quantidade de soluções utilizada para o experimento (Pop), a melhor

solução encontrada (Mín), a quantidade de vezes que esta solução ocorreu (Vezez), o tempo gasto pelo algoritmo (em segundos) e a média das melhores soluções encontradas (Média).

Tabela 2. Melhores resultados obtidos com o SA. (Autoria própria)

Problema	Pop	Mín	Vezez	T(s)	Média
1	50	2951,2	1	426,9	3381,7
2	75	2740	1	680,6	3082,6
3	100	2134,9	1	446,6	2362,13
4	75	1979	1	403,8	2155,3
5	100	1351	1	271,4	1446,63
6	25	1256	1	60,4	1372
7	24	672,7	9	16,8	672,84
8	24	606	8	17,94	606,3
9	14	438,3	10	4,33	438,3
10	14	364	10	4,43	364
11	12	344,9	10	2,94	344,9
12	12	305	10	2,92	305

Com o objetivo de avaliar a eficiência do SA, foram extraídos os melhores resultados, de todos os experimentos computacionais do algoritmo. Considerando o tempo de execução, pode-se concluir que o método obtém resultados satisfatórios, mesmo com problemas com um número maior de cidades. Observando o número de vezes que o algoritmo alcança a melhor solução, nas instâncias maiores, nota-se certa deficiência em manter estabilidade de convergência para o local ótimo no espaço de busca, na medida em que aumenta o número de cidades. Do problema 01 ao 06, o SA encontra apenas uma vez o mínimo, já nos problemas 07 e 08, alcançam 9 e 8 vezes, respectivamente, o ótimo global, e nos problemas de 9 a 12, o resultado global é encontrado nas 10 execuções. Com relação a qualidade das soluções, as tabelas abaixo tratam dos comparativos entre os resultados do SA e os trabalhos [1, 2 e 3].

Na Tabela 3, são apresentados os melhores resultados do SA comparados aos obtidos pelo Método Exato. Em [1], foram abordadas apenas 3 instâncias do problema: 12, 7 e 6 cidades e não definiu-se os tempos de execução do algoritmo. Em [3] também podemos encontrar os mesmos resultados, diferindo de [1] por tratar dos 12 problemas (em todas as instâncias), e adicionar aos resultados o tempo de execução do algoritmo. Por esses motivos, utilizou-se neste trabalho os dados de [3]. Para cada problema são designadas duas colunas apresentando, nesta ordem: a melhor solução encontrada (Mín) e o tempo gasto pelo algoritmo (em segundos).

Tabela 3. SA versus Método Exato. (Autoria própria)

Problema	Exato		SA	
	Mín	T(s)	Mín	T(s)
1	1942,3	6737,7	2951,2	426,9
2	1973	5708,5	2740	680,6
3	1719,2	86325,3	2134,9	446,6
4	1676	85204	1979	403,8
5	1339,9	85439	1351	271,4
6	1223	73266	1256	60,4
7	672,7	0,1	672,7	16,8
8	606	0,1	606	17,94
9	438,3	0	438,3	4,33
10	364	0	364	4,43
11	344,9	0	344,9	2,94
12	305	0	305	2,92

Na maioria dos problemas, o SA obteve aproximação do local globalmente ótimo, com exceção dos problemas 07 ao 12, que retornou a solução exata. Observa-se que, em problemas maiores, com maior número de cidades, o tempo de processamento atingiu um máximo de 680,6 segundos, ocorrido no Problema 02, enquanto a solução exata executou com 5708,5 segundos o mesmo problema. De modo geral, o SA obteve resultados dentro do esperado se considerarmos os valores das soluções mínimas encontrado e o tempo de execução para problemas com o número de cidades maior.

Na Tabela 4, são apresentados os melhores resultados do SA comparados aos obtidos pelo Algoritmo Genético Tradicional (AGT) em [2], que utiliza apenas instâncias menores, por isso a comparação somente com

os problemas de 7 à 12. Para cada problema são designadas quatro colunas apresentando, nesta ordem: a melhor solução encontrada (Mín), a quantidade de vezes que esta solução ocorreu (Vezez), o tempo gasto pelo algoritmo (em segundos) e a média das melhores soluções encontradas (Média).

Tabela 4. SA versus AGT. (Autoria própria)

Problema	Pop	SA				AGT			
		Mín	Vezez	T(s)	Média	Mín	Vezez	T(s)	Média
7	96	672,7	10	69,97	672,7	672,7	2	26,1012	689,96
8	96	606	10	70,29	606	606	4	26,0636	609,8
9	14	438,3	10	4,33	438,3	438,3	19	1,91965	438,7
10	14	364	10	4,43	364	364	20	2,0319	364
11	12	344,9	10	2,94	344,9	344,9	17	2,0584	345,41
12	12	305	10	2,92	305	305	20	2,15975	305

Nestes problemas, os dois métodos utilizaram a mesma quantidade de soluções e foram executados utilizando a plataforma Scilab. O SA foi executado 10 vezes e o AGT 20 vezes. O SA apresentou-se com bom desempenho em termos de processamento computacional e qualidade das soluções, encontrando o local globalmente ótimo em todas as execuções. A afirmação pode ser comprovada analisando as médias dos experimentos e a quantidade de vezes em que o AGT encontrou o local globalmente ótimo, em uma quantidade inferior ao SA.

Na Tabela 5, são apresentados os melhores resultados Algoritmo Memético (AM) e o resultado obtido com o SA para a mesma quantidade de soluções do AM. Para cada problema são designadas três colunas apresentando, nesta ordem: a melhor solução encontrada (Mín), a quantidade de vezes que esta solução ocorreu (Vezez) e o tempo gasto pelo algoritmo (em segundos).

Tabela 5. SA versus AM. (Autoria própria)

Problema	Pop	SA			AM		
		Mín	Vezez	T(s)	Mín	Vezez	T(s)
1	100	3014,6	1	913,5	2070,8	1	31,2
2	75	2740	1	680,6	2154	1	12,2
3	100	2134,9	1	446,6	1705,6	1	13,1
4	100	2005	1	554,8	1668	1	20,9
5	100	1351	1	271,4	1321	5	8,7
6	50	1284	1	129,5	1223	5	3,7
7	75	672,7	10	53,3	672,7	20	1,5
8	100	606	10	70,29	606	20	3,3
9	25	438,3	10	8,49	438,3	20	0,5
10	25	364	10	8,91	364	20	0,5
11	25	344,9	10	6,01	344,9	20	0,4
12	25	305	10	5,76	305	20	0,4

Nos problemas de 1 a 4, o SA não obteve uma boa aproximação do local globalmente ótimo, para os problemas 5 e 6, obteve aproximação e nos problemas 07 ao 12, retornou a mesma solução encontrada pelo AM. Observa-se que, com relação a quantidade de vezes em que os métodos retornaram a solução globalmente ótima, as duas metaheurísticas o mesmo resultado em todas as iterações, o SA foi executado 10 vezes e o AM 20 vezes. Com relação ao tempo de processamento do algoritmo, o SA obteve resultados equiparados a qualidade das soluções visto que os experimentos foram executados em processadores e plataformas diferentes o AM com a utilização da plataforma C++ e o SA no Scilab. De modo geral, o AM mostrou-se mais eficiente que o SA, porém, cabe salientar que a estratégia utilizada em [3], parte do AG na forma tradicional, alterando a taxa dos operadores genéticos, que inclui a mutação em sua operação e recebe um único valor para a execução do crossover e mutação e, a execução do operador de busca local, estratégias que aceleram e refinam a busca no espaço de soluções.

Na Tabela 6, são apresentados alguns experimentos realizados com o SA, alterando o parâmetro T_i (temperatura inicial) de 100 para 500.

Tabela 6. $T_i=100$ versus $T_i=500$. (Autoria própria)

Problema	Pop	Ti =100				Ti =500			
		Mín	Veze	T(s)	Média	Mín	Veze	T(s)	Média
3	25	2350,1	1	126,3	2577	2104	1	148,9	2344,2
4	25	2188	1	133,9	2292,8	2104	1	140	2255,5
5	25	1447,9	1	60,9	1508,8	1443	1	74	1523
3	50	2356,4	1	252,9	2517	2040,5	1	297,7	2285,8
4	50	1351,9	1	122,3	1443,1	1351,4	1	141,7	1459,6
6	50	1284	1	129,5	1376,1	1241	1	151,4	1373,6
5	75	1372,8	1	169,7	1470,5	1360	1	205,2	1429,4
6	75	1279	1	193,7	1324,9	1223	1	226,1	1343,9
6	100	1269	1	265,1	1316,4	1228	1	299,5	1313,6

Conforme [9], a qualidade da solução obtida pelo SA depende, sobretudo, da temperatura inicial. Uma temperatura muito alta, associada a um esquema de resfriamento muito lento, pode conduzir a excelentes soluções porém, com um tempo de processamento bastante elevado. Por outro lado, uma temperatura baixa tende a consumir menos tempo de processamento, porém, pode gerar soluções bastante pobres. Portanto, o parâmetro de temperatura inicial é, por si só, um objeto de experimentações para escolha da melhor configuração inicial. O comportamento típico do algoritmo é aceitar grande diversificação no início, quando a temperatura está alta. À medida que ela decresce, poucas soluções piores são aceitas, uma vez que diminui a probabilidade de se aceitar movimentos de piora, pois, quando $T \rightarrow 0$, $e^{-\Delta/T} \rightarrow 0$ e assim a busca em uma determinada vizinhança é intensificada. Pode-se comprovar analisando os resultados da Tabela 6, onde com uma temperatura inicial maior, melhores resultados foram alcançados.

5. CONSIDERAÇÕES FINAIS

O Problema do Caixeiro Viajante (PCV) é um problema clássico e com diversos trabalhos que o utilizam como base para estudos e resolução de problemas similares que tratam de otimização combinatória. Suas variantes estão presentes em inúmeras aplicações práticas. Nesta perspectiva, surge a demanda de elaboração, modelagem, implementações e validações de métodos para sua resolução. Com a análise comparativa, o *Simulated Annealing* constitui-se de um método de obtenção de boas soluções para problemas de otimização difíceis, tanto em relação a qualidade das soluções obtidas quanto ao tempo de execução.

Os objetivos do trabalho foram atingidos, a implementação da metaheurística adaptada ao PCV, pode contribuir com o problema relatado no estudo de caso, ao encontrar soluções globalmente ótimas em alguns casos e próximas do ótimo em outros, com tempos de execução viáveis. Uma característica observada diante da comparação entre os resultados, demonstra que a convergência para solução globalmente ótima, tem certa dependência dos ajustes dos parâmetros e que, consequentemente, determina a capacidade de exploração de soluções mínimas, ou seja, os algoritmos apresentam maior sensibilidade em relação as configurações. Para o problema em questão, foi exemplificado que o parâmetro de temperatura inicial é, por si só, um objeto de experimentações para escolha da melhor configuração inicial.

As metaheurísticas têm uma forte caracterização pelas várias possibilidades e flexibilidade de serem desenvolvidas e aprimoradas, diante disso, sugerimos como trabalhos futuros implementar novas estratégias de geração da solução inicial, da busca na vizinhança e de adequação de todos os parâmetros estabelecidos para o executar o algoritmo, com o objetivo de constatar a eficiência e portanto, a competitividade do método. Pode-se combinar a estratégia de mais de uma metaheurística ou hibridizar um modelo aproximativo com um exato.

6. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] SILVA, G. L. S. Roteamento de veículos nas entregas realizadas por empresa de laticínios: Estudo de caso. 52 f. Monografia (Graduação em Ciência e Tecnologia) – Universidade Federal Rural do Semi-Árido. Angicos, 2012.
- [2] BEZERRA, T. L. A. Algoritmo Genético Aplicado ao Problema de Roteamento de Veículos nas Entregas Realizadas por uma Empresa de Laticínios: Estudo de Caso. 53 f. Monografia (Graduação em Ciência e Tecnologia) – Universidade Federal Rural do Semi-Árido, Angicos. 2013.
- [3] FERNANDES, F. R. S. Metaheurísticas Evolutivas aplicadas ao Problema de Roteamento de Veículos em uma empresa de laticínios no interior do RN: uma abordagem via algoritmos genético e memético. 85 f. Monografia (Graduação em Sistemas de Informação) – Universidade Federal Rural do Semi-Árido. Angicos, 2016.
- [4] KANDA, J. Y.; et al. *Selection of algorithms to solve traveling salesman problems using meta-learning. International Journal of Hybrid Intelligent Systems*, vol. 8, no. 3, p. 117–128, 2011.

-
- [5] GOLDBARG, M. C.; LUNA, H. P. Otimização combinatória e programação linear: modelos e algoritmos. 2^a ed. Rio de Janeiro: Editora Elsevier, 2005.
- [6] KARP, R. M. *Reducibility among combinatorial problems*. Springer US, 1972.
- [7] SOUZA, M. J. F. Inteligência Computacional para Otimização. Universidade Federal de Ouro Preto, Ouro Preto, MG. 2008.
- [8] METROPOLIS, N.; et al. *Equation of State Calculations by Fast Computing Machines*. *The Journal of Chemical Physics*, , 21(6): 1087-1092, 1953.
- [9] KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. *Optimization by simulated annealing*. *Science*, 220(4598):671–680, 1983.
- [10] BECCENERI, J. C. Meta-heurísticas e Otimização Combinatória: Aplicações em Problemas Ambientais. Disponível em: <http://www.lac.inpe.br/elac13/arquivos/MiniCurso_02ELAC2012.pdf>. Acesso em: 02/03/2018.
- [11] FREDERICK S. H.; GERALD J. L. Introdução à pesquisa operacional. Tradução: Ariovaldo Griesi; revisão técnica João Chang Junior. Título original: *Introduction operations research*. São Paulo: McGraw-Hill, 2006.
- [12] BLUM, C.; et al. *Hybrid metaheuristic in combinatorial optimization: a survey*. *Applied Soft Computing*, v. 11, p. 4135–4151. 2011
- [13] KRIPKA, R. M. L.; KRIPKA, M. Simulated Annealing Aplicado na Otimização da Alocação de Salas em Instituição de Ensino Superior. In: XXXI *Iberian-Latin-American Congress on Computational Methods in Engineering*. Buenos Aires/Argentina. 2010
- [14] CERNY, V., *Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm*. *J. Opt. Theory Appl.*, 45, 1, 41-51, 1985.



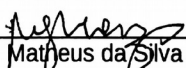
MINISTÉRIO DA EDUCAÇÃO
Universidade Federal Rural do Semi-Árido – UFERSA
Centro de Ciências Exatas e Naturais – CCEN

ATA DE DEFESA DE TRABALHO DE CONCLUSÃO DE CURSO

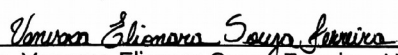
Às 16:30 horas do dia catorze de setembro de dois mil e dezoito, na sala 3 da Central de Aulas V do CCEN – Centro de Ciências Exatas e Naturais da UFERSA – Universidade Federal Rural do Semi-Árido, reuniu-se a Banca Examinadora de defesa de trabalho de conclusão de curso de autoria da aluna **Maria Clara de Oliveira Gê**, aluno do curso de Bacharelado em Ciência e Tecnologia desta universidade, N° de matrícula **2015021050**, com o título **"SIMULATED ANNEALING APLICADO AO PROBLEMA DO CAIXEIRO VIAJANTE: UM ESTUDO DE CASO"**. A Banca Examinadora ficou assim constituída por três membros: Prof. Dr. Matheus da Silva Menezes, presidente da banca e orientador do Trabalho de Conclusão de Curso; Co-Orientadora MSc. Vanessa Elionara Souza Ferreira; Prof. Dr. Leonardo Casillo e Profa. Dra. Danielle Casillo, como membros. Concluída a defesa, procedeu-se o julgamento pelos membros da banca examinadora, tendo a aluna obtido as seguintes notas: 10,0; 10,0; 10,0; 10,0. Apuradas as notas verificou-se que o aluno foi aprovado com média geral 10,0. E para constar, eu, Matheus da Silva Menezes, lavrei a presente ata que, após lida e aprovada pelos membros da banca examinadora, será assinada por todos.

Mossoró, 14 de setembro de 2018.

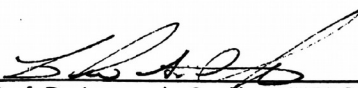
Assinatura dos membros da Banca Examinadora.



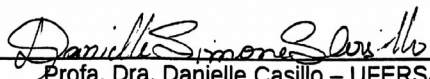
Prof. Dr. Matheus da Silva Menezes – UFERSA
Presidente e orientador



MSc. Vanessa Elionara Souza Ferreira – UFERSA
Co-Orientadora



Prof. Dr. Leonardo Casillo – UFERSA
Primeiro Membro



Profa. Dra. Danielle Casillo – UFERSA
Segundo Membro