

UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE – UERN

FACULDADE DE CIÊNCIAS EXATAS E NATURAIS – FANAT

DEPARTAMENTO DE INFORMÁTICA – DI

Lucas Luís Rebouças Silva

Uma Metaheurística BRKGA com Infecção Viral para o Problema do Caixeiro Viajante

MOSSORÓ – RN

2019

Lucas Luís Rebouças Silva

Uma Metaheurística BRKGA com Infecção Viral para o Problema do Caixeiro Viajante

Monografia apresentada à Universidade do Estado do Rio Grande do Norte como um dos pré-requisitos para obtenção do grau de bacharel em Ciência da Computação, sob orientação do Prof. Dr. Dario José Aloise.

MOSSORÓ – RN

2019

© Todos os direitos estão reservados a Universidade do Estado do Rio Grande do Norte. O conteúdo desta obra é de inteira responsabilidade do(a) autor(a), sendo o mesmo, passível de sanções administrativas ou penais, caso sejam infringidas as leis que regulamentam a Propriedade Intelectual, respectivamente, Patentes: Lei nº 9.279/1996 e Direitos Autorais: Lei nº 9.610/1998. A mesma poderá servir de base literária para novas pesquisas, desde que a obra e seu(a) respectivo(a) autor(a) sejam devidamente citados e mencionados os seus créditos bibliográficos.

Catálogo da Publicação na Fonte.
Universidade do Estado do Rio Grande do Norte.

S586m Silva, Lucas Luís Rebouças
Uma Metaheurística BRKGA com Infecção Viral para o
Problema do Caixeiro Viajante. / Lucas Luís Rebouças
Silva. - Mossoró, 2019.
43p.

Orientador(a): Prof. Dr. Dario José Aloise.
Monografia (Graduação em Ciência da Computação).
Universidade do Estado do Rio Grande do Norte.

1. Ciência da Computação. 2. Otimização
Combinatória. 3. PCV. 4. BRKGA. 5. Infecção Viral. I.
Aloise, Dario José. II. Universidade do Estado do Rio
Grande do Norte. III. Título.

O serviço de Geração Automática de Ficha Catalográfica para Trabalhos de Conclusão de Curso (TCC's) foi desenvolvido pela Diretoria de Informatização (DINF), sob orientação dos bibliotecários do SIB-UERN, para ser adaptado às necessidades da comunidade acadêmica UERN.

LUCAS LUÍS REBOUÇAS SILVA

Uma Metaheurística BRKGA com Infecção Viral para o Problema do Caixeiro Viajante

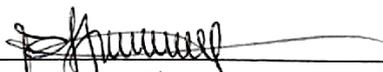
Monografia apresentada como pré-requisito para a obtenção do título de Bacharel em Ciência da Computação da Universidade do Estado do Rio Grande do Norte – UERN, submetida à aprovação da banca examinadora composta pelos seguintes membros:

Aprovada em: 23/09/2019

Banca Examinadora



Prof. Dr. Dário José Aloise
Universidade do Estado do Rio Grande do Norte - UERN



Prof. Dr. Francisco Chagas de Lima Júnior
Universidade do Estado do Rio Grande do Norte - UERN



Prof. Dr. Carlos Heitor Pereira Liberalino
Universidade do Estado do Rio Grande do Norte - UERN

À minha família, em especial aos meus pais e meus avós, os quais são os pilares da minha vida.

AGRADECIMENTOS

Agradeço primeiramente a Deus, com Sua infinita misericórdia, pela força a mim concedida para concluir este trabalho.

À minha família, em especial ao meu pai Jair e minha mãe Patrícia, pela confiança em minha capacidade, pelo amor, paciência, e por prestar todo apoio possível que um pai e uma mãe podem dar.

Agradeço à minha noiva Rebeca, pela paciência, carinho e amor, e por sempre me incentivar.

Ao professor Dr. Dario Aloise, pelas orientações, paciência e por não desistir de mim.

Ao meu amigo Alan Douglas, pela contribuição neste trabalho, auxiliando na implementação de um dos algoritmos.

Um agradecimento especial a Jonathan Lopes, que não mediu esforços para me orientar na implementação dos algoritmos, sempre paciente e prestativo.

A todos que fizeram parte deste ciclo da minha vida, muito obrigado!

*E sabemos que todas as coisas contribuem juntamente
para o bem daqueles que amam a Deus, daqueles
que são chamados segundo o seu propósito.*

Romanos 8:28

RESUMO

Este trabalho apresenta uma metaheurística híbrida composta de um Algoritmo Genético com Chaves Aleatórias Viciadas BRKGA (Biased Random-Key Genetic Algorithm) acrescido da estratégia de melhoria Infecção Viral (BRKGA+IV) para o Problema do Caixeiro Viajante, estratégia ainda não abordada na literatura. A eficácia da metaheurística é comprovada pela comparação dos resultados computacionais obtidos com os das metaheurística Algoritmo Genético Puro (AG), Algoritmo Genético com Infecção Viral (AG+IV), BRKGA e a metaheurística híbrida (BRKGA + IV), aplicado em instâncias testes disponibilizadas na TSPLIB.

Palavras-chave: Otimização Combinatória, Problema do Caixeiro Viajante, BRKGA, Infecção Viral, TSPLIB.

ABSTRACT

This paper presents a hybrid metaheuristic composed of a Biased Random-Key Genetic Algorithm (BRKGA) with the Viral Infection (BRKGA + IV) improvement strategy, a strategy not yet addressed in the literature. The effectiveness of metaheuristics is proven by comparing the computational results obtained with those of the metaheuristics Pure Genetic Algorithm (AG), Viral Infection Genetic Algorithm (AG + IV), BRKGA and hybrid metaheuristic (BRKGA + IV), applied in available test instances. at TSPLIB.

Keywords: Combinatorial Optimization, Traveling Salesman Problem, BRKGA, Viral Infection, TSPLIB.

LISTA DE SIGLAS

PCV	Problema do Caixeiro Viajante
AE	Algoritmos Evolutivos
AG	Algoritmo Genético
RKGA	<i>Randon-Key Genetic Algorithm</i>
BRKGA	<i>Biased Random-Keys Genetic Algorithm</i>
IV	Infecção Viral
TSPLIB	Library of Traveling Salesman Problems

LISTA DE FIGURAS

Figura 2.1 - Instância do PCV representado por um grafo	20
Figura 2.2 - Restrições de Cardinalidade para o PCV	22
Figura 3.1 - Etapas do Algoritmo Genético.....	27
Figura 3.2 - Processo evolutivo do RKGA.....	30
Figura 3.3 - Passos do BRKGA.....	32
Figura 4.1 - Exemplo de um vírus para um determinado cromossomo	33
Figura 4.2 - Passos do BRKGA com a Infecção Viral	35
Figura 5.1 - Representação de uma matriz de distâncias para o PCV	36
Figura 5.2 - Representação de um Cromossomo.....	37
Figura 5.3 - Processo de Transcrição	38
Figura 6.1 - Gráfico que representa a porcentagem de melhora das soluções do BRKGA + IV em relação ao BRKGA genérico	43
Figura 6.2 - Gráfico de Comparação de Resultados	43

LISTA DE TABELAS

Tabela 2.1 - Tempo para calcular a melhor rota em função de n cidades.....	20
Tabela 6.1 - Resultados do BRKGA.....	39
Tabela 6.2 - Resultados do BRKGA + IV	40
Tabela 6.3 – Comparativo entre os Resultados do BRKGA e BRKGA+IV	41
Tabela 6.4 - Comparativo de Resultados com o ótimo de cada instância	42

SUMÁRIO

1	INTRODUÇÃO	17
1.1	Otimização Combinatória	17
1.2	Metaheurísticas	17
1.3	Metaheurísticas Híbridas	18
1.4	Desenvolvimento do Trabalho	18
2	PROBLEMA DO CAIXEIRO VIAJANTE	19
2.1	Definição	19
2.2	Formulações Matemáticas	21
2.3	Aplicações	23
3	MÉTODOS DE PESQUISA	24
3.1	Algoritmos Evolutivos	25
3.2	Algoritmos Genéticos	26
3.3	Algoritmos Genéticos com Chaves Aleatórias (RKGA)	29
3.4	Algoritmos Genéticos com Chaves Aleatórias Viciadas (BRKGA)	31
4	INFECÇÃO VIRAL	33
4.1	O Vírus	33
4.2	Taxa de Infectividade	34
4.3	A Infecção	34
5	BRKGA COM INFECÇÃO VIRAL APLICADO AO PCV	35
5.1	Representação do Problema	35
5.2	Função de Aptidão	36
5.3	Decodificador	36
5.4	Infecção Viral Aplicada ao BRKGA	37
6	RESULTADOS COMPUTACIONAIS	39
7	CONCLUSÃO E TRABALHOS FUTUROS	44
	REFERÊNCIAS	45

1 INTRODUÇÃO

1.1 Otimização Combinatória

Problemas de Otimização Combinatória visam minimizar ou maximizar uma função objetivo, sujeito a um conjunto de soluções viáveis, conjunto este, discreto. A complexidade computacional destes problemas aumenta conforme o tamanho do conjunto de soluções viáveis. Geralmente, problemas de otimização combinatória são classificados como NP-Difíceis, não existindo algoritmos polinomiais capazes de obter soluções ótimas (a menos que $P = NP$), o que acaba atraindo diversos pesquisadores de diferentes áreas (GAREY e JOHNSON, 1979). Através destes problemas, é possível resolver muitas questões práticas. O problema a ser utilizado neste trabalho será o Problema do Caixeiro Viajante (PCV). O PCV consiste em encontrar o ciclo hamiltoniano de menor custo, onde, cada vértice em um dado grafo, só deve ser visitado uma única vez (exceto o vértice que é tanto a origem quanto o destino). O PCV é um dos problemas mais populares (HOFFMAN e WOLFE, 1985); (MELAMED et al., 1990) sendo muito utilizado em diversas áreas tais como: Indústrias (através da minimização do custo por meio da melhor rota obtida por equipamentos no processo de montagem), empresas (transportadoras, entrega e coleta de cargas), transporte de passageiros e etc.

1.2 Metaheurísticas

O PCV contém vários métodos de resolução, através de abordagens Exatas, Heurísticas ou Metaheurísticas. Os métodos exatos, em muitos casos, exigem um grande esforço computacional, sendo assim inviável na resolução de problemas que contenham um grande número de vértices (cidades), devido ao espaço de busca aumentar exponencialmente. A abordagem Heurística resolve o PCV através dos métodos mais conhecidos: *Busca Tabu* por Hansen (1986) e Glover (1990); *Simulated Annealing (SA)* por Kirkpatrick et al. (1983); *Greedy Randomized Adaptive Search Procedures (GRASP)* introduzido por Feo e Resende (1995), entregando normalmente, boas soluções (aproximativas) em tempo aceitável. A abordagem Metaheurística pode ser definida como um método heurístico para resolver problemas de otimização combinatória. Geralmente são aplicadas para problemas

NP-Difíceis, onde não se conhece um algoritmo eficiente para resolução destes problemas. No presente trabalho, a metaheurística empregada será a evolutiva, que busca através de algoritmos evolutivos (baseado na seleção natural proposta por Charles Dawin) soluções aproximativas em tempo polinomial.

1.3 Metaheurísticas Híbridas

Metaheurística Híbrida é um termo utilizado para definir a combinação entre uma determinada metaheurística com outra técnica de otimização. A hibridização visa combinar as vantagens de cada técnica, retornando melhorias em termos de velocidade ou precisão computacional.

Os AGs utilizam operadores de reprodução, cruzamento e mutação nos cromossomos (possível solução). A partir destas operações, novas populações são geradas, através da combinação de melhores soluções progressivamente aprimoradas. Baseado nos AGs, os Algoritmos Genéticos com Chaves Aleatórias (RKGA) foram introduzidos por Bean (1994) tendo como característica a representação dos cromossomos através de vetores no intervalo $[0,1]$ e a utilização das partições Elite, Não-Elite e Mutante. O Algoritmo Genético com Chaves Aleatórias Viciadas (BRKGA) é uma evolução do RKGA, diferenciando apenas na forma em que o pai Elite é selecionado. Segundo (GONÇALVES, 2012), o BRKGA apresenta melhor desempenho quando se trata de tempo de execução. No trabalho proposto, utilizaremos um BRKGA junto a técnica da Infecção Viral (IV), o que constitui uma metaheurística híbrida.

1.4 Desenvolvimento do Trabalho

O presente trabalho propõe um comparativo entre duas abordagens da metaheurística BRKGA: a genérica e a acrescida de uma Infecção Viral (IV), ambas aplicadas ao PCV. Será organizado da seguinte forma: no capítulo 2 é feita uma apresentação detalhada sobre o PCV, onde será exposto as formulações matemáticas e as principais aplicações do problema. No capítulo 3 serão apresentados as características e os algoritmos em que as metaheurísticas são baseadas. No capítulo 4 é feito um estudo sobre infecção viral. No capítulo 5, explicamos as metaheurísticas e comparamos seus resultados em relação ao

problema proposto. No capítulo 6 serão exibidos os resultados computacionais. Finalmente, no capítulo 7 será apresentada a conclusão.

2 PROBLEMA DO CAIXEIRO VIAJANTE

Problemas de otimização combinatória sempre atraiu diversos pesquisadores; matemáticos, engenheiros, cientistas da computação entre outros (GUEDES, 2005). Esses problemas são modelados para resolver questões na vida real. Um exemplo disso é o Problema do Caixeiro Viajante (PCV), que pode ser utilizado, por exemplo, na otimização de rotas em uma empresa de distribuição, com o objetivo de reduzir o tempo da viagem, conseqüentemente, cortando custos.

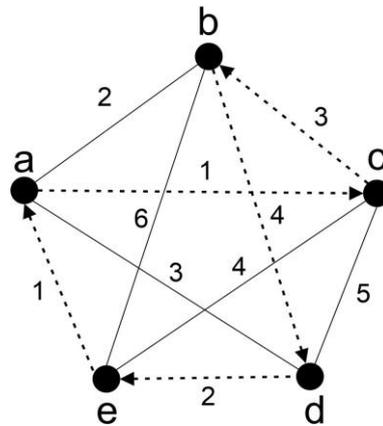
2.1 Definição

O PCV consiste em encontrar o ciclo hamiltoniano de menor custo. Pode ser representado através de um grafo composto de N vértices (cidades), onde, entre cada par de cidades existe um custo. O objetivo é encontrar entre as $(n-1)!$ possibilidades, aquela que passa por todas as cidades e que seja de menor custo. A restrição imposta ao caixeiro é de que ele não pode passar mais de uma vez por uma mesma cidade e deve encerrar sua viagem na cidade pela qual iniciou o percurso. As soluções viáveis são as possíveis rotas que o caixeiro viajante pode seguir. Os custos associados a estas soluções são as distâncias percorridas entre as cidades.

O PCV pode ser classificado como simétrico ou assimétrico. Simétrico quando, para todos os pares de nós $\{i, j\}$ possui os custos c_{ij} e c_{ji} iguais e assimétrico quando os custos c_{ij} e c_{ji} são diferentes.

A figura 2.1 mostra um exemplo de instância do PCV com cinco cidades ABCDE, representado por um grafo $G(V,E)$, onde cada vértice V representa uma cidade e cada arco E representa um caminho entre um par de cidades. Um exemplo de viagem seria ABCDEA com custo total 12, onde o custo é a soma dos pesos dos arcos em sequência. A solução ótima para a instância em questão seria: ACBDEA com custo total 11.

Figura 2.1 - Instância do PCV representado por um grafo



Fonte: Autoria Própria

Segundo Goldberg e Luna (2005), a importância do PCV está vinculada a pelo menos três características: *Grande aplicação prática* (Bellmore e Nemhauser [1968], Burkard [1979], Reinelt [1994]), *uma enorme relação com outros modelos* (Laporte et al. [1996]), e por sua *grande dificuldade de solução exata* (Papadimitriou e Steiglitz [1982], Gould [1991] e Zhang [1997]).

O PVC é considerado um problema de classe NP-Difícil, sendo assim intratável por métodos exatos. O PVC possui grande dificuldade de solução exata, devido a sua característica combinatória.

A tabela 2.1 mostra o tamanho do esforço computacional ao listar todas as soluções possíveis e entre elas selecionar a melhor, utilizando um computador capaz de processar 1 bilhão de adições por segundo.

Tabela 2.1 - Tempo para calcular a melhor rota em função de n cidades

n	Rotas por segundo	$(n-1)!$	Tempo de resposta
5	250 milhões	24	insignificante
10	110 milhões	362 880	0,003 s
15	71 milhões	87 bilhões	20 min
20	53 milhões	$1,2 \times 10^{17}$	73 anos
25	42 milhões	$6,2 \times 10^{23}$	87 milhões de anos

Fonte: Adaptado de Guedes, Leite & Aloise (2005)

Para resolução de problemas como este, técnicas heurísticas e metaheurísticas são utilizadas, onde, diferente dos métodos exatos, buscam uma

solução aproximativa, dispondo de um tempo menor (em alguns casos, muito menor) de execução, além de reduzir consideravelmente o uso de hardware.

2.2 Formulações Matemáticas

O PCV possui diversas formulações matemáticas, todas elas têm como objetivo principal a eliminação de sub-rotas, com isso, sendo possível fugir do ótimo local e buscar o ótimo global. Nesta subseção, citaremos 2 (duas) bastante difundidas.

- Formulação de Dantzig-Fulkerson-Johnson (DFJ)

Dantzig, Fulkerson e Johnson (1954) propuseram uma formulação para o PCV, dado um grafo $G = (N, A)$, onde N é o conjunto de nós ou vértices, e A é o conjunto de arcos ou arestas que conectam cada par de cidades i e $j \in N$:

$$(PCV1) \text{ Minimizar } Z = \sum_{j=1}^n \sum_{i=1}^n c_{ij} x_{ij}$$

Sujeito a:

$$\sum_{i=1}^n x_{ij} = 1 \quad \forall j \in N \quad (2.1)$$

$$\sum_{j=1}^n x_{ij} = 1 \quad \forall i \in N \quad (2.2)$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1 \quad \forall S \subset N \quad (2.3)$$

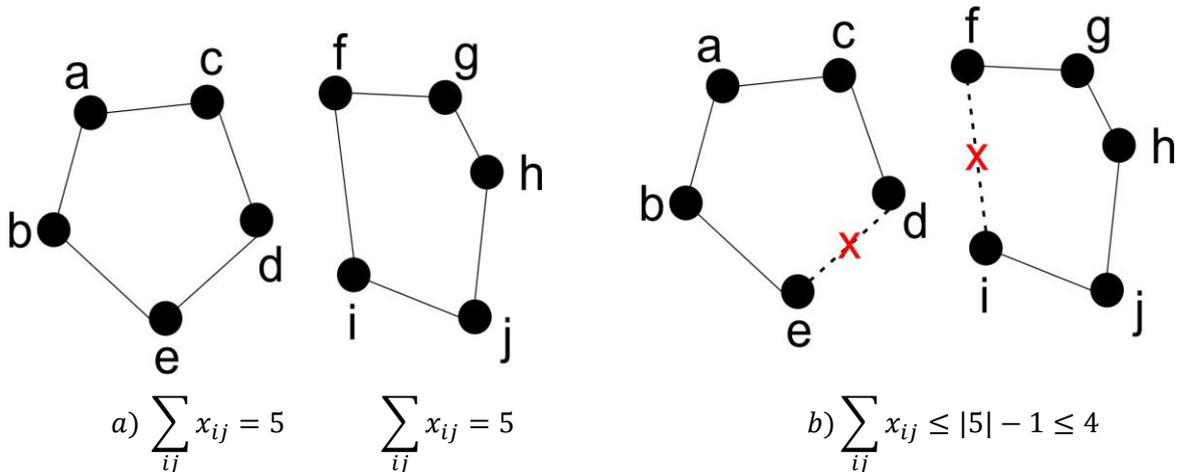
$$x_{ij} \in \{0,1\} \quad \forall i,j \in N \quad (2.4)$$

Onde a variável binária x_{ij} assume um valor igual a 1, caso o arco $(i,j) \in A$ pertencer a uma solução, e 0 em caso contrário, e S é um subgrafo de G , onde $|S|$ representa o número de vértices. Nessa formulação, supõe-se que x_{ii} não existe e, que teremos $n(n-1)$ variáveis inteiras 0-1 e $O(2^n)$ restrições. O conjunto de restrições

(2.3) elimina as chances de ocorrer sub-rotas (rota que não inclui todas as n cidades). As restrições (2.1) e (2.2) são semelhantes às do problema de designação.

De acordo com a figura 2.2, o conjunto de restrições apresentadas são capazes de eliminar a ocorrência de sub-rotas. As equações em $|S|$ tornam esses circuitos ilegais.

Figura 2.2 - Restrições de Cardinalidade para o PCV: (a) Solução Ilegal; (b) Restrições Associadas



Fonte: Autoria Própria

Essa formulação é importante, pois mostra a natureza combinatória do PCV.

- Formulação de Miller-Tucker-Zemlin

Outra formulação para o PCV foi proposta por Miller, Tucker e Zemlin (1960), denominada Folha de Cravo.

Inicialmente é necessário definir uma cidade origem. O viajante deve visitar as outras $n-1$ cidades exatamente uma vez. Durante o percurso deve retornar à cidade origem exatamente t vezes, incluindo o retorno final, e não deve visitar mais de p cidades diferentes em um ciclo.

Para garantir a existência de rotas viáveis, a formulação necessita que $\left\lceil \frac{n-1}{p} \right\rceil \leq t \leq n-1$, onde $\lceil a \rceil$ denota o menor inteiro maior ou igual ao valor de a , ou o teto da divisão.

$$(PCV2) \text{ Minimizar } Z = \sum_{j=1}^n \sum_{i=1}^n c_{ij} x_{ij}$$

Sujeito a:

$$\sum_{i=2}^n x_{i1} = t \quad (2.5)$$

$$\sum_{i=1}^n x_{ji} = 1 \quad j = 2, \dots, n \quad (2.6)$$

$$\sum_{j=1}^n x_{ji} = 1 \quad i = 2, \dots, n \quad (2.7)$$

$$u_i - u_j + px_{ij} \leq p - 1 \quad 2 \leq i \neq j \leq n \quad (2.8)$$

$$u_i \geq 0 \quad 2 \leq i \leq n \quad (2.9)$$

$$x_{ij} \in \{0,1\} \quad \forall i, j \in N \quad (2.10)$$

O primeiro conjunto de restrições de (PCV2) dá a garantia de que a cidade origem é visitada exatamente t vezes. Esta formulação envolve $O(n^2)$ restrições em $n^2 - 1$ variáveis. Quando $t = 1$ e $p \geq n - 1$ o PCV será modelado.

2.3 Aplicações

O PCV está presente em inúmeros problemas práticos. A seguir algumas aplicações encontradas na literatura sobre o PCV:

- Determinação de acesso a arquivos computacionais (Henry-Labordere [1969]);
- Programação de assistência social (Saskena [1970]);
- Manipulação de itens em estoque (Ratliff e Rosenthal [1981]);
- Na maioria dos problemas de roteamento de veículos (Bodin [1983] e de navios Fagerholt e Christiansen [2000]);
- Programação de transporte entre células de manufatura (Finke e Kusiak [1985]);
- Programação de operações de máquinas em manufatura (Kusiak e Finke [1987]);
- Otimização do movimento de ferramentas de corte (Chauny et al. [1987]);
- Otimização de palhetas em turbinas a gás (pante et al. [1987]);

- Análise de estruturas cristalinas (Bland e Shallcross [1987]);
- Fabricação de Chips VLSI (Korte [1988]);
- Otimização de perfurações de furos em placas de circuitos impressos (Reinelt [1989]);
- Na solução de problemas de sequenciamento de tarefas (Whitley [1991]);
- Trabalhos administrativos (Laporte et al. [1996]);
- Roteamento de entrega postal (Laporte et al. [1996]);
- Na solução de problemas de programação e distribuição de tarefas em plantas (Salomon et al.[1997]);
- Cortes em chapas de aço e vidro (Hoeft e Palekar [1997]);
- Construção de árvore evolucionária e determinação de sequenciamento genético (Korostensky e Gonnet [2000]);
- Produção de Aço (Tang et al. [2000]);
- Testes citológicos (Laporte e Palekar [2002]);
- Planejamento da produção (Ben-Arieh et al. [2003]);
- Planejamento de sistemas noturnos de distribuição de valores (Cordone e Calvo [2003]);
- Programação de tripulação de companhias aéreas (Barnhart et al. [2003]).

3 MÉTODOS DE PESQUISA

Métodos exatos são utilizados quando precisamos de resultados precisos. Muitas vezes isso não é possível, devido à complexidade de alguns problemas e seus altos custos computacionais. Tal complexidade pode ser reduzida utilizando técnicas, tais como a do paralelismo, ou seja, múltiplos processadores trabalhando simultaneamente a fim de gerar um resultado em tempo satisfatório. Porém, nem sempre dispomos de máquinas suficientes para tal. Pensando nisso, foram desenvolvidos métodos heurísticos, visando uma menor utilização de hardware aliado a uma execução em tempo aceitável. Em alguns problemas, como o do PCV, por apresentarem característica combinatória, são difíceis de obter uma solução exata. Isso acontece devido ao crescimento exponencial das possibilidades em relação ao tamanho do problema. Métodos Heurísticos surgem como uma possível solução, apesar de não garantirem uma solução ótima, esses métodos buscam

soluções aproximativas. A Metaheurística por sua vez, é uma evolução da heurística. Geralmente são aplicadas para problemas NP-Difíceis, onde não se conhece um algoritmo eficiente para resolução destes problemas. Foi desenvolvida visando oferecer uma maior eficiência, entregando uma garantia maior de encontrar um resultado ótimo (solução viável) ou o mais próximo possível do ótimo. São capazes de se adaptar a qualquer problema, geralmente de origem combinatória.

3.1 Algoritmos Evolutivos

De acordo com Menezes (2014), um Algoritmo Evolutivo (AE) é um termo utilizado para a programação evolutiva, estratégias evolutivas e algoritmos genéticos. Atualmente, os AEs formam uma classe de algoritmos muito estudada, com inúmeras variações e aplicações bem sucedidas na resolução de problemas em várias áreas da ciência (MICHALEWICZ, 1996; MICHALEWICZ e FOGEL, 2000, apud RIBEIRO FILHO, G, 2001).

Os AEs baseiam-se em métodos naturais já conhecidos. Exemplos deles são os algoritmos genéticos e suas variações, que modelam computacionalmente a teoria da seleção natural proposta por Charles Darwin (1858). Essa teoria diz que na natureza todos os indivíduos competem entre si por recursos de sobrevivência, os indivíduos mais fracos têm sua descendência reduzida, enquanto os mais fortes estão aptos a gerar um número maior de descendentes.

São utilizados operadores genéticos, como cruzamento e mutação, e cada indivíduo recebe uma avaliação (fitness). Baseados nesta avaliação são aplicados operadores genéticos, a fim de simular a sobrevivência do mais apto. Os algoritmos evolutivos buscam (dentro da atual população) as soluções que possuem as melhores características, e através do cruzamento, tenta gerar soluções cada vez melhores. O processo se repete até que se obtenha uma solução satisfatória em um tempo suficientemente aceitável.

Os Algoritmos evolutivos dependem de fatores probabilísticos na fase de inicialização da população e na fase final de evolução, com isso, pode-se afirmar que nem sempre será possível obter os melhores resultados. Para conseguir resultados superiores, se faz necessário utilizar algumas técnicas, tais como: busca local, redes neurais, infecção viral entre outras.

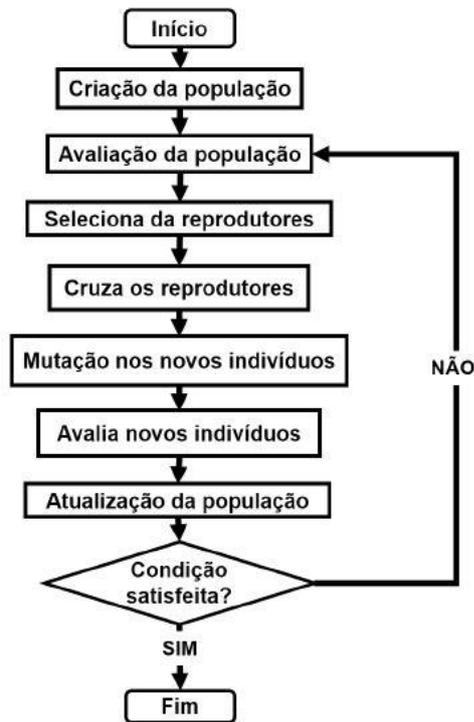
3.2 Algoritmos Genéticos

Os Algoritmos Genéticos (AG) são uma subárea dos algoritmos evolutivos, sendo assim, baseados na teoria da seleção natural. John H. Holland (1975) foi o responsável por introduzir a ideia de simular matematicamente os fenômenos de adaptação (naturais ou artificiais), incluindo estes mecanismos para um modelo computacional. “A técnica, consiste em explorar o espaço de busca através de uma população de soluções viáveis (cromossomos), que evolui a cada geração por meio de operadores genéticos de cruzamento e mutação.” (GUEDES, 2005). Os cromossomos representam uma rota, no caso do PCV, onde um conjunto de 1 a 8 cidades, {3, 8, 7, 2, 1, 5, 4, 6} pode representar uma das soluções viáveis. Devido sua grande capacidade de alteração e adaptação a diversos problemas de otimização, essa técnica vem sendo muito utilizada (OLIVEIRA, 2015).

Segundo Konzen et al. (2003), o AG utiliza parâmetros resultantes da seleção natural. Estes parâmetros são conhecidos como operadores genéticos de cruzamento (crossover) e mutação, e possuem a função de manipular indivíduos de uma população, por meio de gerações, com objetivo de melhorar e adaptar (fitness) a mesma.

Na resolução de um problema, os algoritmos genéticos seguem etapas baseadas no processo evolutivo, como o apresentado na figura 3.1.

Figura 3.1 - Etapas do Algoritmo Genético



Fonte: Adaptado de Holland (1992)

Os Algoritmos Genéticos possuem alguns conceitos fundamentais:

- Indivíduos

Conceito importante em AGs, indivíduos são conhecidos como cromossomos e suas características como genes. A decodificação dos genes gera o fenótipo (valor de um indivíduo). Cada indivíduo possui um grau de adaptação, medido por uma função objetivo. Um grau de aptidão (fitness) é atribuído a cada indivíduo com o propósito de informar o nível de adaptação do mesmo em relação aos demais (GOLDBERG, 1989).

- População

O processo evolutivo da população acontece por meio da seleção, avaliação, cruzamento e mutação. Uma geração representa o número de vezes que a população passou pela execução do processo de seleção, reprodução, mutação e atualização (DAVIS, 1991).

- Inicialização

A população inicial é gerada de forma randômica, visando uma maior variedade biológica, fazendo com que o espaço de busca seja ampliado (GOLDBERG, 1989).

- Avaliação

É responsável por determinar a probabilidade de o indivíduo transmitir os seus genes para a próxima geração (LIMA, 2008). Cada indivíduo é avaliado para que o grau de adaptação seja verificado, analisando a importância do mesmo na resolução de um problema proposto.

- Seleção

Etapa onde a adaptação dos indivíduos é avaliada, julgando se suas características serão propagadas por gerações. Sua implementação é baseada no processo de seleção natural, onde os indivíduos mais aptos possuem maior probabilidade de gerar mais descendentes, enquanto que os menos aptos poderão ainda gerar descendentes, porém em menor número.

- Reprodução

Posterior à etapa de Seleção, na reprodução os indivíduos seguem com uma probabilidade pré-definida pelo processo de cruzamento (*crossover*), combinando parte dos genes dos pais para gerações de filhos. Os indivíduos geralmente são escolhidos de forma aleatória.

- Mutação

Nesta etapa, o algoritmo simula o fenômeno natural da mutação genética de indivíduos, modificando o valor de alguns genes de forma

aleatória. A mutação acontece nos indivíduos resultantes do cruzamento e com uma probabilidade pré-definida de efetuar algum tipo de alteração em sua estrutura (GOLDBERG, 1989). Após sofrer uma mutação, um cromossomo passa a buscar áreas que ainda não foram exploradas, com isso é possível diversificar a população e fugir de ótimos locais.

- Atualização

Inserção dos indivíduos resultantes do cruzamento e mutação na população, sobrepondo aos que lhe deram origem. Outra alternativa é a substituição de indivíduos apenas quando os mesmos forem mais aptos, ou seja, quando possuírem melhor valor *fitness* (DAVIS, 1991).

- Finalização

É um teste realizado com o propósito de descobrir se o algoritmo genético conseguiu atingir a condição de parada. Um dos critérios que pode ser utilizado é o número de gerações evoluídas.

3.3 Algoritmos Genéticos com Chaves Aleatórias (RKGA)

Bean (1994) foi o introdutor dos Algoritmos Genéticos com Chaves Aleatórias (RKGA – *Random-Key Genetic Algorithm*), pensando em tratar problemas clássicos de sequenciamento e otimização.

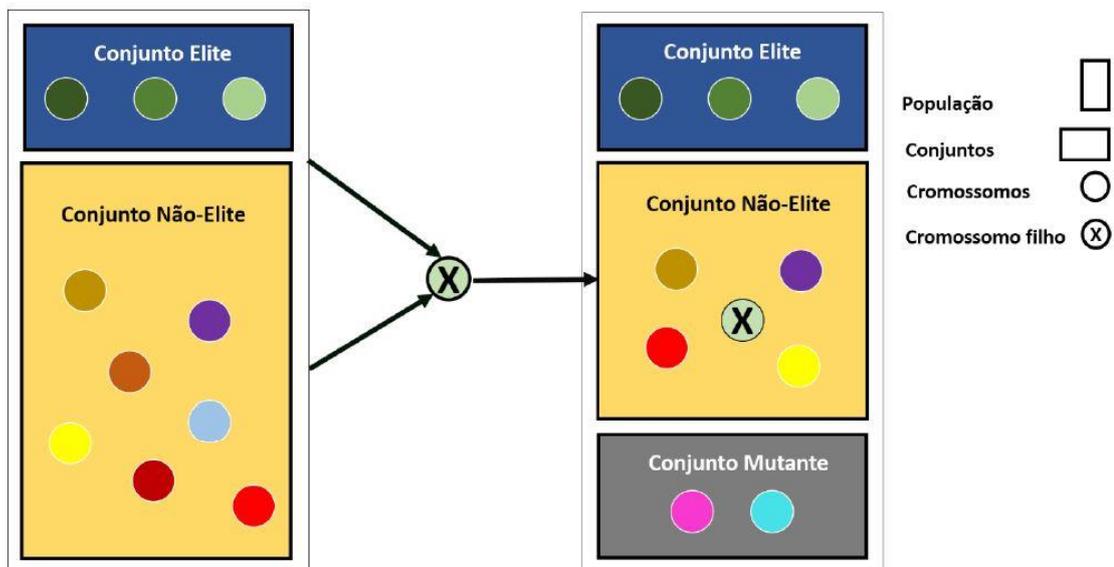
Normalmente, um operador de cruzamento (realiza a troca de genes entre dois cromossomos) como o *cruzamento de um ponto (one point crossover)*, pode gerar proles (resultado de um cruzamento entre dois cromossomos-pai) onde não apresentam uma solução viável para um determinado problema. Visto isso, Bean (1994) propôs um algoritmo que utiliza chaves aleatórias geradas no intervalo de $[0,1]^n$, onde os cromossomos são representados como vetores de números reais gerado aleatoriamente neste intervalo, e n representa o número de genes em um cromossomo. Um algoritmo determinístico, chamado *decodificador*, utiliza como entrada um vetor solução e associa a ele uma solução do problema de otimização combinatória. Um decodificador tem como objetivo mapear um vetor de chaves

aleatórias na solução de um problema de otimização, calculando o custo desta solução.

O RKGA é um algoritmo evolutivo, aplicando assim o princípio darwinista, onde os indivíduos mais fortes estão mais aptos a propagar o seu material genético. Neste algoritmo, um novo cromossomo é originado a partir de dois cromossomos, tendo este novo, uma maior probabilidade de herdar características do cromossomo mais apto. Em cada geração, a população atual é dividida em três grupos diferentes: *Elite*, *Não-elite* e *Mutante*. O conjunto Elite apresenta os melhores valores para a resolução de um problema, o conjunto Não-elite formam o restante da população, enquanto os mutantes são responsáveis por não permitir que a população convirja para um ótimo local.

De acordo com a figura 3.2, o conjunto elite é copiado para próximas gerações, de diferente forma, o conjunto dos mutantes tem seus cromossomos substituídos por novos, gerados aleatoriamente. Os cromossomos do conjunto Não-elite são substituídos por novos cromossomos (filhos), obtidos através do cruzamento de dois cromossomos (pais) escolhidos de modo aleatório.

Figura 3.2 - Processo evolutivo do RKGA



Fonte: Adaptado de Resende (2013)

O RKGA é um método flexível, ou seja, é capaz de adaptar-se a diversos tipos de problemas, tais como: programação de tarefas em máquinas em paralelo, caixeiro viajante, roteamento de veículos, problema da mochila, cobertura de conjuntos, entre outros.

3.4 Algoritmos Genéticos com Chaves Aleatórias Viciadas (BRKGA)

O BRKGA é uma metaheurística evolutiva baseada no RKGA de Bean (1994) para resolver problemas de otimização discreta e global, onde a solução é representada por um vetor de n chaves aleatórias.

Segundo Resende (2013), o BRKGA é constituído de dois elementos característicos: As *chaves aleatórias*: Um número real gerado aleatoriamente no intervalo $[0,1]$; e o *decodificador*: Responsável por mapear um vetor de chaves aleatórias em uma determinada solução para um problema de otimização, possuindo a capacidade de calcular o seu custo apenas ordenando o vetor de chaves, gerando assim uma permutação correspondente aos índices dos elementos ordenados.

O BRKGA possui como aspectos característicos: ser um método aleatório; utiliza estratégia elitista onde os filhos gerados herdaram mais características do pai elite; não existe mutação nos cruzamentos.

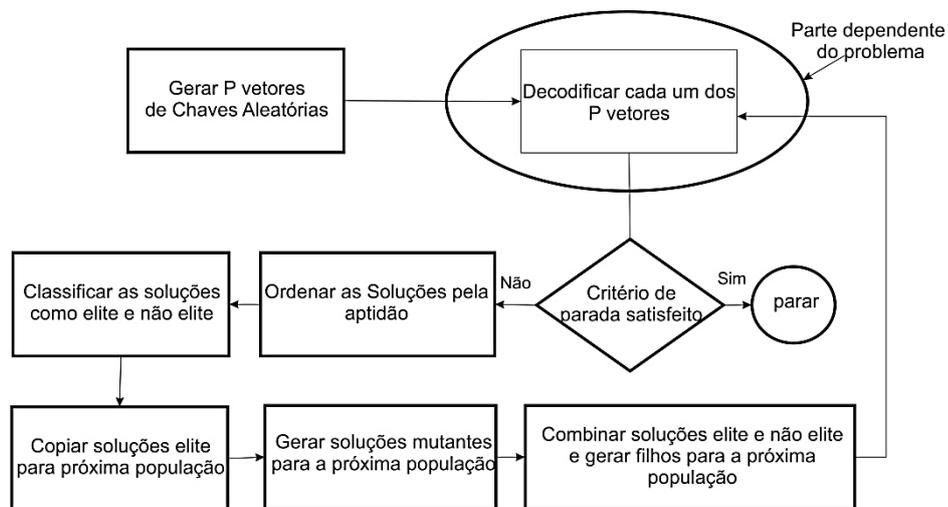
Assim como no RKGA, o princípio darwinista é utilizado no BRKGA, onde os indivíduos mais aptos de uma população possuem mais chances de se reproduzir e perpetuar seu material genético.

O algoritmo é iniciado com uma população de p vetores e n chaves aleatórias, produzindo assim várias populações. Seguindo os passos citados abaixo:

- a) Na k -ésima geração, os p vetores da população são divididos em um pequeno conjunto de $p_e < p/2$ vetores correspondentes às melhores soluções. Esse conjunto é chamado de *elite* enquanto o outro conjunto com o restante da população é chamado de *não-elite*;
- b) Todos os vetores *elite* são copiados, sem modificação, para a população da $k+1$ -ésima geração, conceito de elitismo;
- c) Posteriormente, p_m vetores de chaves aleatórias são inseridos na população da $k+1$ -ésima geração. Esses vetores são identificados como *mutantes*, e assim como no RKGA, tem como objetivo impedir que a população convirja para um ótimo local;
- d) Para completar os p elementos da população da $k+1$ -ésima geração, $p-p_e-p_m$ vetores são gerados combinando pares de soluções da população da k -ésima geração com a combinação uniforme parametrizada de Spears e DeJong (1991), onde apenas uma solução é *elite*, enquanto a outra, *não-elite*.

No BRKGA existe uma divisão entre a parte genética da solução e a parte específica de cada problema. Isso faz com que seja mais fácil adaptar o algoritmo a diferentes problemas. De acordo com a figura 3.3, a parte independente não possui conhecimento algum sobre a parte dependente do problema. O decodificador é o único intermediador entre ambas, produzindo soluções a partir dos vetores de chaves aleatórias e calculando os valores das respectivas aptidões.

Figura 3.3 - Passos do BRKGA. A figura circulada representa a parte dependente do problema, as demais figuras representam a parte independente.



Fonte: Adaptado de Gonçalves e Resende (2011)

O BRKGA funciona praticamente igual ao RKGA em relação ao cruzamento e mutação. A diferença principal entre ambos está na seleção dos pais para o cruzamento. No caso, o BRKGA seleciona um cromossomo-pai do conjunto Elite e um do conjunto Não-elite, de forma aleatória, enquanto o RKGA seleciona os pais de toda a população. Essa diferença faz com que indivíduos mais aptos tenham maior probabilidade de transmitir seus genes para as próximas gerações. No RKGA cada pai pode ser o pai *a* ou o pai *b*, no BRKGA o pai *a* é sempre o pai elite enquanto o pai *b* sempre será não-elite.

Os algoritmos baseados no BRKGA encontram soluções melhores que nos baseados no RKGA, tendo como unidade comparativa o tempo de execução (GONÇALVES, 2012).

4 INFEÇÃO VIRAL

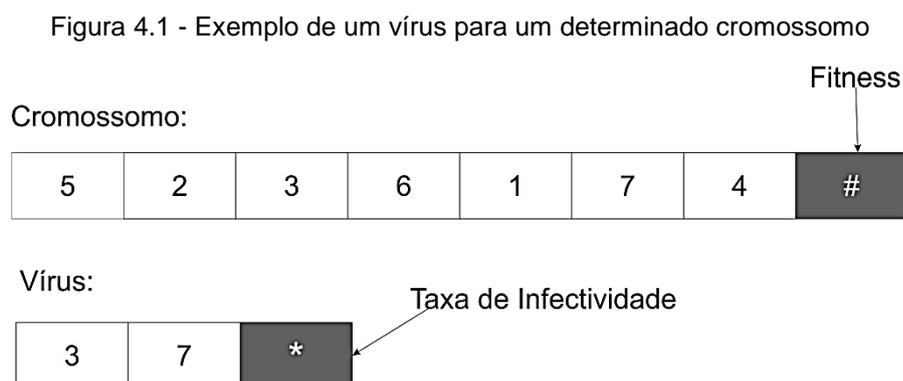
De acordo com a teoria da evolução, o material genético pode ser transferido de um organismo para outro através de um vírus. Esse vírus pode infectar um organismo e, através desta infecção, reescrever o gene do organismo que foi infectado (NAKAHARA E SAGAWA, 1986; KANOH, 1996).

Autores como Kanoh & Nakamura (2000), Kanoh et al. (1996), Kanoh, Kondo & Sugimoto (2002), Guedes, Leite & Aloise (2005) em seus estudos, utilizam a infecção viral juntamente com o Algoritmo Genético, substituindo o operador de mutação pelo operador de infecção. Esse método fundamenta-se em manter uma população adicional, conhecida como População de Vírus, contendo partes de soluções viáveis do problema. Segundo Kanoh, Kondo & Sugimoto (2002), o uso da infecção viral no Algoritmo Genético melhora a taxa de busca do algoritmo.

4.1 O Vírus

Um vírus é tido como uma solução parcial e tem como característica infectar um organismo (cromossomo) reescrevendo o material genético do mesmo, podendo aumentar ou diminuir o seu valor de *fitness*. O vírus é selecionado da mesma forma que o cromossomo-pai, mas ao invés de possuir um *fitness*, é atribuído a ele um valor chamado de *infectividade*, o qual é uma espécie de indicador de qualidade do vírus.

No caso do problema do caixeiro viajante, onde o cromossomo representa uma rota completa, um vírus vai corresponder a uma parte da rota, de acordo com a figura 4.1.



Fonte: Adaptado de Fontes, Soares, Aloise, Silva, Oliveira (2007).

Um vírus é um hospedeiro, sendo assim, necessita de organismo para interagir. Nos algoritmos evolutivos um vírus só poderá ser atuante na evolução de uma geração se existir um agente no qual ele possa hospedar-se.

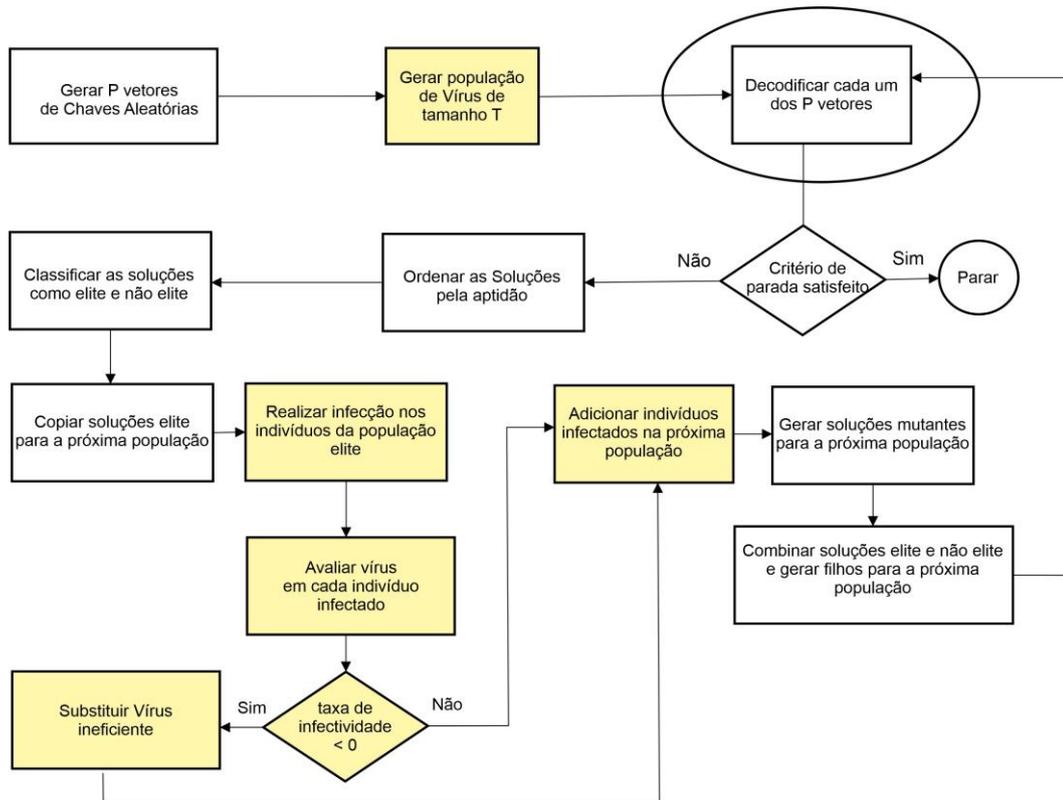
4.2 Taxa de Infectividade

Indica o quanto o vírus pode melhorar uma determinada solução. Todos os vírus possuem inicialmente uma taxa de infectividade igual, podendo aumentar até um valor máximo ou podendo reduzir a zero. O valor inicial e o limite máximo dessa taxa são previamente estipulados. Quando a infecção causa uma melhora da *fitness* do cromossomo, o valor da infectividade é acrescido em 1 (dependendo do limite pré-estabelecido pelo algoritmo), caso contrário, esse valor é decrementado até chegar a 0.

4.3 A Infecção

O vírus é selecionado para infectar um indivíduo de acordo com sua infectividade. Um vírus com uma maior taxa de infectividade é mais propenso infectar um indivíduo do que um vírus com uma baixa taxa de infectividade (LARTIGUE). O ato de infectar um indivíduo é chamado de transcrição. Esse processo *“consiste em o vírus modificar o cromossomo infectado de forma que esse contenha um trecho idêntico ao representado pelo agente infectante e fazer os ajustes necessários para a manutenção da viabilidade do indivíduo receptor da carga viral”* (GUEDES, LEITE E ALOISE, 2005).

Figura 4.2 - Passos do BRKGA com a Infecção Viral



Fonte: Autoria Própria

5 BRKGA COM INFECÇÃO VIRAL APLICADO AO PCV

Neste capítulo serão abordados alguns métodos evolutivos de resolução para o PCV. Nas implementações a seguir, usaremos o BRKGA genérico e o BRKGA acrescido da Infecção Viral para resolução das instâncias do PCV.

5.1 Representação do Problema

Os elementos do problema, como a matriz de distâncias, as chaves aleatórias, a população de cromossomos, população de vírus, as cidades e a distância entre cada uma delas, foram representados através de vetores uni e bidimensionais que armazenam todas as informações importantes, como por exemplo, cromossomos, genes, populações, etc. Cada cromossomo é um vetor unidimensional que pode conter a possível solução do problema.

A tabela 5.1 demonstra uma forma de representar o problema através de uma matriz de adjacência, contendo as distâncias entre as cidades. Esta foi a representação utilizada neste trabalho.

Figura 5.1 - Representação de uma matriz de distâncias para o PCV

CIDADES	1	2	3	4	5	6	7
1	0	42	61	30	17	82	31
2	42	0	14	87	28	70	19
3	61	14	0	20	81	21	8
4	30	87	20	0	34	33	91
5	17	28	81	34	0	41	34
6	82	70	21	33	41	0	19
7	31	19	8	91	34	19	0

Fonte: Autoria Própria

5.2 Função de Aptidão

Uma solução é viável se a partir de uma cidade, todas as demais cidades sejam percorridas, sem repetição, retornando à cidade de origem.

A função de aptidão é responsável por determinar o quão bom é uma solução, conferindo uma nota para cada cromossomo de acordo com o problema. Esta função verifica, dentre a população de cromossomos, o cromossomo (rota) que possui a melhor fitness, ou seja, a rota onde se encontra a menor distância. No AG convencional, a função de aptidão precisa verificar se um cromossomo é válido, colocando-o na lista de cromossomos aptos a gerar uma boa solução, porém, no BRKGA essa verificação não se faz necessária, visto que cada chave aleatória possui um ID (ou posição) atrelado a ela, fazendo com que a permutação desses IDs forme uma rota válida (falaremos um pouco mais sobre esse processo no próximo subtópico).

5.3 Decodificador

No algoritmo proposto, temos os processos de Codificação e Decodificação das chaves aleatórias. O processo de geração de uma chave aleatória é denominado Codificação. Já na Decodificação, um Decodificador é responsável por mapear as chaves aleatórias, transformando-as em uma solução real do problema. O Decodificador é utilizado para ordenar a solução, conforme a tupla *posição/chave*, gerando uma permutação nas posições (onde cada posição é representada por uma cidade), caracterizando assim, a solução para o problema em questão. Após o

processo de decodificação, a solução pode ser avaliada para definir sua respectiva aptidão. O processo de Codificação e Decodificação é ilustrado na figura 5.2.

Figura 5.2 - Representação de um Cromossomo (a) Codificado e (b) Decodificado

a) Cromossomo Codificado

POS = 1	POS = 2	POS = 3	POS = 4	POS = 5
Chave: 0,34	Chave: 0,76	Chave: 0,12	Chave: 0,78	Chave: 0,11

b) Cromossomo após Decodificação (Ordenado)

POS = 5	POS = 3	POS = 1	POS = 2	POS = 4
Chave: 0,11	Chave: 0,12	Chave: 0,24	Chave: 0,76	Chave: 0,78

Fonte: Autoria Própria

5.4 Infecção Viral Aplicada ao BRKGA

Apesar de o BRKGA genérico apresentar boas soluções para o problema proposto, neste trabalho, adicionamos uma infecção viral para buscar soluções ainda melhores, visto que esta técnica faz com que o espaço de busca seja expandido.

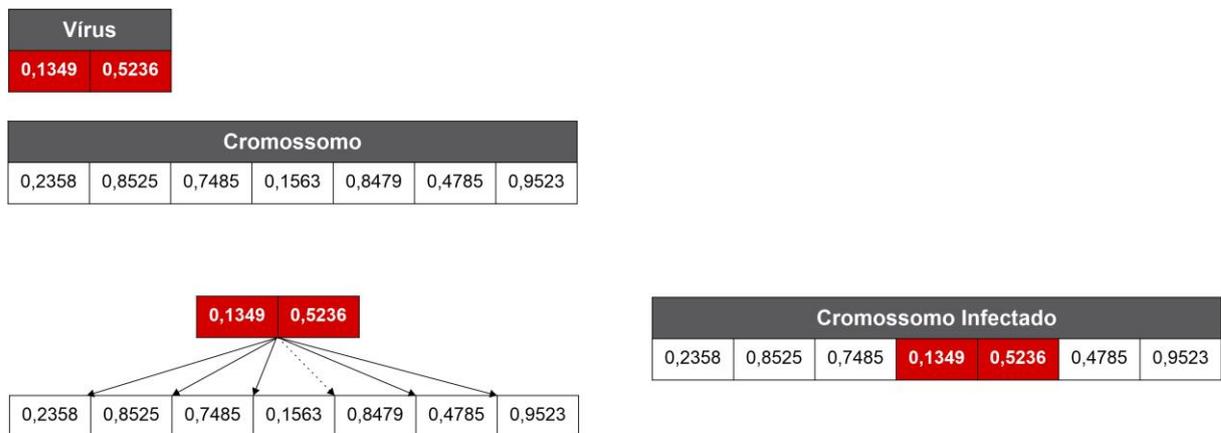
A infecção irá ocorrer em metade da população elite, onde cada indivíduo será infectado por todos os vírus da população, em seguida, será verificado se o vírus em questão beneficiou este indivíduo, melhorando sua aptidão. Caso haja uma melhora, o vírus sobrescreve os genes do indivíduo infectado, que então será copiado para próxima geração. O algoritmo 1 demonstra o processo da infecção.

Algoritmo 1 – Infecção Viral

1	Infecção
2	Selecione os melhores indivíduos
3	Para cada um desses indivíduos faça
4	Transcrição com cada vírus
5	Se indivíduo aumentou o fitness
6	Taxa de infectividade do vírus aumenta
7	Senão
8	Taxa de infectividade do vírus diminui
9	Fim se
10	Aceita-se os X melhores indivíduos da população
11	Fim Para
12	Fim

No processo da Transcrição (figura 5.3), o vírus é testado entre cada par de genes do indivíduo, sendo escolhida a posição onde ele melhor beneficiar este indivíduo. No presente trabalho, os vírus foram gerados aleatoriamente.

Figura 5.3 - Processo de Transcrição



Fonte: Adaptado de Fontes, Soares, Aloise, Silva, Oliveira (2007).

Como falado anteriormente, cada vírus possui uma taxa de infectividade, a qual define o quanto o vírus foi benéfico em um determinado indivíduo. Quando o vírus é criado ou modificado, sua infectividade é definida em um valor fixo (neste trabalho, definimos a taxa inicial como 2). Quando a infecção traz benefícios ao indivíduo, esta taxa é acrescida em 1, porém, se um determinado vírus não beneficiar o indivíduo, esta taxa é decrementada em 1. Chegando a taxa em 0, o vírus será excluído e outro será gerado em seu lugar, voltando a ter taxa de infectividade igual a inicial. Este processo ocorre a cada infecção em um indivíduo.

Como isso, a partir de cada iteração, a população de vírus tende a ficar com melhores trechos de soluções.

6 RESULTADOS COMPUTACIONAIS

Para desenvolver a metaheurística proposta, utilizamos a linguagem JAVA. Os testes foram realizados em um Desktop com processador Intel Pentium G4560 de 3.5 Ghz e 8GB de memória RAM no Sistema Operacional Windows 10.

Para fazer uma comparação entre os métodos, foram realizados diversos testes em 13 instâncias, 3 simétricas e 10 assimétricas. Inicialmente, foram utilizadas 10 instâncias conforme Guedes, Leite & Aloise (2005) e posteriormente foram adicionadas 3 novas instâncias retiradas da TSPLIB (biblioteca de instâncias para o PCV), verificando a qualidade média das soluções obtidas, o desvio padrão e o tempo médio (Tabelas 6.1 e 6.2).

Foram realizados 20 testes em cada instância, com o critério de parada definido em 3000 iterações para cada um dos algoritmos (BRKGA e BRKGA+IV).

O BRKGA foi configurado da seguinte maneira:

- População $P = 200$,
- População-Elite $P_e = 20\%$ da população P .
- Vício Probabilístico = 0.7
- População-Filhos = 70% da população P
- População-Mutante = 10% da população P .

Tabela 6.1 - Resultados do BRKGA

Instâncias	Média	Desvio Padrão	Tempo (s)
br17.atsp	39	0	0,71
ftv170.atsp	7320,5	355,367	5,91
ftv33.atsp	1564,3	96,305	1,25
ftv64.atsp	2705,2	118,220	2,24
ftv70.atsp	2911,4	146,508	2,60
kro124p.atsp	54994,2	3161,132	4,49
p43.atsp	5685,1	18,136	1,73

brazil58.tsp	32131,85	2010,967	1,98
ry48p.atsp	16625,1	824,479	1,70
rbg323.atsp	2425,8	39,429	13,91
rbg403.atsp	3644,4	59,788	17,47
brg180.tsp	42780	9071,752	6,70
swiss42.tsp	1465	97,500	1,61

Fonte: Autoria Própria

No processo da infecção, metade da população-elite foi infectada de forma aleatória, e os novos indivíduos gerados são postos em uma nova população denominada infectados.

O BRKGA configurado para Infecção Viral:

- População $P = 200$,
- População-Elite $P_e = 20\%$ da população P ,
- População-Filhos = 40% da população P ,
- População-Infectados = 30% da população P ,
- População-Mutante = 10% da população P .

Os parâmetros do vírus foram configurados da seguinte forma:

- Tamanho do Vírus = 5% do tamanho do indivíduo,
- Taxa de Infectividade inicial = 2,
- Taxa de Infectividade mínima = 0;

Tabela 6.2 - Resultados do BRKGA + IV

Instâncias	Média	Desvio Padrão	Tempo (s)
br17.atsp	39	0	2,04
ftv170.atsp	4530,9	241,420	221,39
ftv33.atsp	1449,6	56,637	7,91
ftv64.atsp	2172,8	124,110	30,75
ftv70.atsp	2329,5	126,794	37,20
kro124p.atsp	46215,25	2034,065	73,74
p43.atsp	5648,9	15,955	12,56

brazil58.tsp	29277,95	1737,270	25,75
ry48p.atsp	16114,05	626,706	16,20
rbg323.atsp	1832,65	106,671	1000,07
rbg403.atsp	2739	51,682	1934,22
brg180.tsp	4522,6	2033,180	251,32
swiss42.tsp	1414,7	73,819	14,33

Fonte: Autoria Própria

Em relação as tabelas 6.1 e 6.2, em algumas instâncias, percebeu-se que o desvio padrão obteve um alto valor. Isso diz respeito a variação no resultado entre cada execução, o que pode ser explicado por o BRKGA ser um algoritmo não-determinístico e aleatório.

A tabela a seguir demonstra que a adição da Infecção Viral no BRKGA melhorou a qualidade das soluções em 12 das 13 instâncias.

Tabela 6.3 – Comparativo entre os Resultados do BRKGA e BRKGA+IV

TSPLIB	BRKGA		BRKGA + IV	
	Tempo Médio (s)	Média de Solução	Tempo Médio (s)	Média de Solução
br17.atsp	0,71	39	1,845	39
ftv170.atsp	5,91	7320	212,396	4530
ftv33.atsp	1,25	1564	6,910	1449
ftv64.atsp	2,24	2705	27,754	2173
ftv70.atsp	2,60	2911	31,201	2329
kro124p.atsp	4,49	54194	73,331	46215
p43.atsp	1,73	5685	12,568	5649
brazil58.tsp	1,97	32131	25,759	29277
ry48p.atsp	1,69	16625	16,201	16114
rbg323.atsp	13,91	2425	994,632	1833
rbg403.atsp	17,47	3644	1835,638	2739
brg180.tsp	6,704	42780	251,328	4522
swiss42.tsp	1,614	1465	14,334	1414

Fonte: Autoria Própria

Pode-se notar de acordo com a tabela 6.3, que instâncias com maiores cidades obtiveram um tempo extra para concluir a execução, sendo que no BRKGA com Infecção Viral, este tempo foi bem maior, porém a melhora nas soluções foi significativa se comparado ao BRKGA genérico. Em contrapartida, instâncias com menos cidades não obtiveram uma melhora relevante, neste caso, o BRKGA seria uma opção mais eficiente se compararmos o tempo de execução e o resultado em si.

Em todas as instâncias, exceto a *br17.atsp*, que manteve o mesmo resultado, o BRKGA com infecção viral melhorou a solução. Em alguns casos, como a instância *ftv170.atsp*, a melhora na solução foi de aproximadamente 38%, já na instância *brg180.tsp*, onde o BRKGA obteve um baixo desempenho, quando adicionado a infecção viral, a melhora foi de cerca de 89% (figura 6.2). Outra instância que obteve uma melhoria em sua solução foi a *rbg403.atsp*, onde o BRKGA + IV melhorou cerca de 26%, chegando próximo do ótimo, o que reforça a importância da intensificação da busca aplicado por esta técnica.

Tabela 6.4 - Comparativo de Resultados com o ótimo de cada instância

TSPLIB		Média das Soluções			
Instâncias	Ótimo	AG	AG+IV	BRKGA	BRKGA+IV
br17.atsp	39	39	39	39	39
ftv170.atsp	2755	22771	7115	7320	4530
ftv33.atsp	1286	2490	1593	1564	1449
ftv64.atsp	1839	6205	2536	2705	2173
ftv70.atsp	1950	6951	2754	2911	2329
kro124p.atsp	36230	139310	52037	54194	46215
p43.atsp	5620	6205	5668	5685	5649
brazil58.tsp	25395	86647	30475	32131	29277
ry48p.atsp	14422	37650	16629	16625	16114
rbg323.atsp	1326	5421	2342	2425	1833
rbg403.atsp	2465	6830	3564	3644	2739
brg180.tsp	1950	85626	25415	42780	4522
swiss42.tsp	1273	3388	1583	1465	1414

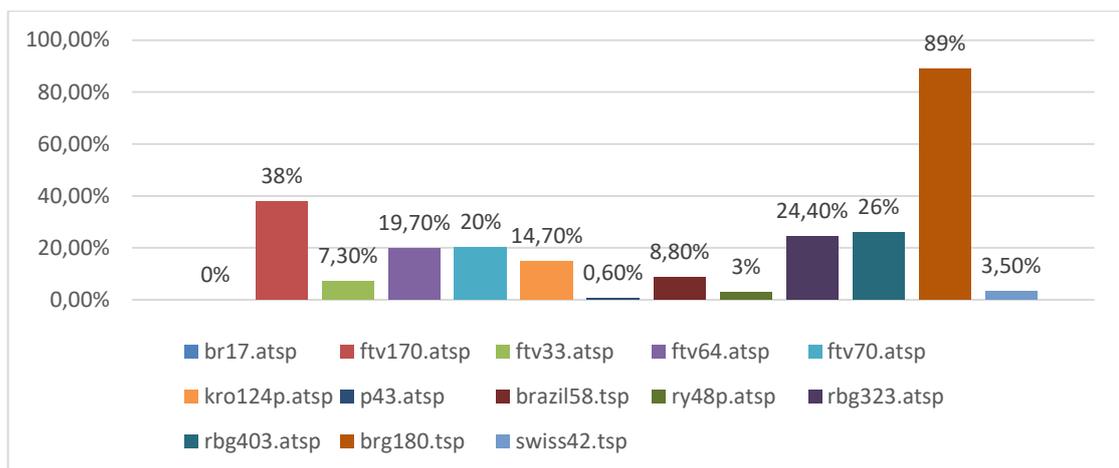
Fonte: Autoria Própria

Na tabela 6.4, é feita uma comparação dos resultados obtidos nos algoritmos evolutivos (AG, AG + IV, BRKGA, BRKGA + IV) com o ótimo de cada instância. Para realizar esta comparação, definimos no AG e no AG + IV parâmetros semelhantes aos demais algoritmos, tais como tamanho da população e número de gerações.

De acordo com a tabela 6.4, pode-se confirmar que BRKGA aliado a Infecção Viral, supera os demais algoritmos em 93% das instâncias, obtendo resultados ótimos ou próximos do ótimo.

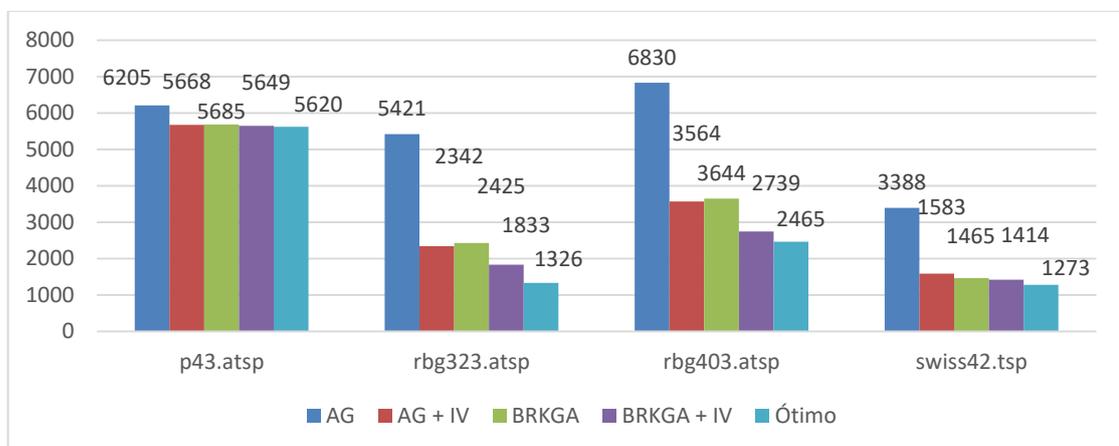
Essa comparação respalda a pesquisa até aqui realizada, onde o BRKGA acrescido da Infecção Viral prova-se uma técnica eficiente em buscar resultados ótimos ou próximos do ótimo.

Figura 6.1 - Gráfico que representa a porcentagem de melhora das soluções do BRKGA + IV em relação ao BRKGA genérico



Fonte: Autoria Própria

Figura 6.2 - Gráfico de Comparação de Resultados



Fonte: Autoria Própria

7 CONCLUSÃO E TRABALHOS FUTUROS

Neste trabalho, apresentou-se uma metaheurística BRKGA junto a técnica da Infecção Viral, como proposta de solução para o Problema do Caixeiro Viajante.

Vários testes computacionais foram realizados, de forma ser possível afirmar que o BRKGA acrescido da Infecção Viral traz melhores resultados se comparado ao BRKGA genérico. Isso é possível devido a característica aleatória do BRKGA, que propõe diversificar a busca (evitando que as soluções analisadas fiquem confinadas a uma pequena região do espaço total), junto a intensificação que a técnica da Infecção Viral traz consigo, que procura melhorar a solução o máximo possível.

Para trabalhos futuros, fica a sugestão de implementação de outras estratégias junto ao BRKGA, como a Busca Local e, na Infecção Viral, a Simbiose.

REFERÊNCIAS

GOLDBARG, M. C. PACCA LUNA, H. *Otimização Combinatória e Programação Linear*. 2.ed. Rio de Janeiro: Elsevier, 2005.

GUEDES, A. C. B.; LEITE, J. N. F.; ALOISE, D. J. *Um algoritmo genético com infecção viral para o problema do caixeiro viajante*. Publica 1, p.16 – 24, 2005.

HOLLAND, J.H. *Adaptation in natural and artificial systems* . Michigan: MIT Press, 1975.

KANO, H. et al. *Solving Constraint Satisfaction Problems by a Genetic Adopting Viral Infection*. In: IEEE. 1996.

KANO, H.; NAKAMURA, N. *Route Guidance with Unspecified Staging Posts Using Genetic Algorithm for Car Navigation Systems*. In *IEEE Intelligent Transportation Systems*. Dearborn (MI), USA. 2000.

KANO, H., KONDO, M.; SUGIMOTO, M. *Solving Timetabling Problems Using Genetic Algorithms Based on Minimizing Conflict Heuristics*. In: CIMNE, Barcelona, 2002.

MICHALEWICZ Z. *Genetic algorithms + data structures = evolution programs*. Berlin: Springer Verlag, 1996. p. 387.

MICHALEWICZ Z.; FOGEL D. B. *How to solve it: modern heuristics*. Berlin: Springer Verlag, 2000. 467p.

RIBEIRO FILHO, G. *Melhoramentos No Algoritmo Genético Construtivo e Novas Aplicações em Problemas de Agrupamento*. 2001. Tese (Doutorado em Computação Aplicada). Instituto Nacional de Pesquisas Espaciais - INPE, São José dos Campos. p. 129.

FONTES, F. F. C.; *Algoritmo Memético com Infecção Viral: Uma Aplicação ao Problema do Caixeiro Viajante Assimétrico*. 72 f. Tese (Mestrado) - Universidade Federal do Rio Grande do Norte, Programa de Engenharia de Produção. Natal, 2006.

MENEZES, M. S.; *O problema do Caixeiro alugador com coleta de prêmios: um estudo algorítmico*. 126 f. Tese (Doutorado) - Universidade Federal do Rio Grande do Norte Centro de Ciências Exatas e da Terra. Programa de Pós-Graduação em Sistemas e Computação. Natal, 2014.

GOLDBERG, D. E. *Genetic algorithms in search, optimization & machine learning*. [S.l.]: Addison-Wesley, 1989. **DAVIS, L. D.** *Handbook of genetic algorithms*. [S.l.]: Van Nostrand Reinhold, 1991.

GEYER-SCHULZ, A. *Fuzzy Rule-Based Expert Systems and Genetic Machine Learning (Studies in Fuzziness and Soft Computing)*. [S.l.]: Physica-Verlag HD, 1996.

RESENDE, M. G. C.; *Introdução aos algoritmos genéticos de chaves aleatórias viciadas*, Anais do Simpósio Brasileiro de Pesquisa Operacional - SBPO, (2013).

OLIVEIRA, T. H. F. *Abordagem híbrida para o problema de atribuição de localidades a anéis em redes SONET/SDH*. Dissertação de Mestrado – UERN/UFERSA, 2015.

BEAN, J. C. *Genetic algorithms and random keys for sequencing and optimization*. ORSA Journal on Computing, v. 6, p. 154–160, 1994.

FONTES, F. F. C. & GUEDES, A. C. B. & SOARES, W.K.S. & ALOISE, D. J. & SILVA, A. C. G. & OLIVEIRA, W. *Uma Proposta de Infecção Viral em Algoritmos Meméticos: Aplicação no Problema do Caixeiro Viajante Assimétrico*. XXVII Encontro Nacional de Engenharia de Produção. Paraná, 2007.

REBOUÇAS, A. D. C. *Desenvolvimento de uma Metaheurística BRKGA com Infecção Viral para o Problema de Atribuição de Localidades a Anéis em Redes SONET/SDH*. 2018. 57 f. TCC (Graduação) - Curso de Ciência da Computação, Departamento de Informática - di, Universidade do Estado do Rio Grande do Norte, Mossoró, 2018.