

FERRAMENTA COMPUTACIONAL PARA ANÁLISE DO COMPORTAMENTO DA POPULAÇÃO DO ALGORITMO GENÉTICO NO ESPAÇO DE SOLUÇÕES

COMPUTATIONAL TOOL FOR ANALYSIS OF THE BEHAVIOR OF GENETIC ALGORITHM POPULATION IN THE SOLUTION SPACE

 Stanley Jefferson de Araujo Lima¹

 Sidnei Alves de Araújo²

Resumo

O Algoritmo Genético (AG) é um método otimização inspirado na teoria de evolução das espécies que tem sido largamente empregado na solução de problemas classificados como *NP-Hard* (*Non-deterministic Polynomial Time*), entre os quais estão o Problema de Sequenciamento da Produção (*Job Shop Scheduling Problem* – JSP) e o Problema de Roteamento de Veículos (PRV). Entretanto, encontrar soluções para qualquer problema de otimização empregando o AG pressupõe a adoção de um esquema de codificação das soluções e a configuração dos operadores genéticos. Diferentes esquemas e configurações podem produzir comportamentos diferentes na população do AG, mas observar tais comportamentos não é uma tarefa fácil e, por este motivo, vem atraindo a atenção de muitos pesquisadores ao longo dos últimos anos. Neste trabalho propõe-se uma ferramenta computacional que permite analisar como o esquema de codificação e os operadores genéticos afetam o comportamento da população do AG no espaço de soluções, por meio de visualização dos indivíduos projetados para um espaço bidimensional e de medidas de desempenho implementadas na ferramenta. Nos experimentos conduzidos analisou-se o comportamento da população do AG em função de três esquemas de codificação de soluções para o Problema de Roteamento de Veículos Capacitados (PRVC). Como resultados, além de uma discussão acerca da análise do comportamento do AG, pode-se constatar que as medidas de desempenho fornecidas pela ferramenta computacional desenvolvida podem auxiliar na proposição e/ou escolha de heurísticas que visem apoiar o processo de refinamento das soluções geradas pelo AG, melhorando o seu desempenho.

Palavras-chave: Algoritmos Genéticos. Codificação de soluções. Espaço de soluções. Problema de Roteamento de Veículos Capacitados.

Abstract

Genetic Algorithm (GA) is an optimization method inspired by the theory of evolution of species that has been widely used to solve problems classified as NP-Hard (Non-deterministic Polynomial Time), among which are the Job Shop Scheduling Problem (JSP) and the Vehicle Routing Problem (VRP). However, finding solutions to any optimization problem using GA presupposes the adoption of a solution encoding scheme and the configuration of genetic operators. Different schemes and configurations may produce different behaviors in the GA population, but observing such behaviors is not an easy task and, for this reason, has attracted the attention of many researchers over the last years. This work proposes a computational tool that allows to analyze how the encoding scheme and genetic operators affect the behavior of the GA population in the solution space, by visualizing the individuals projected to a two-dimensional space and performance measures implemented in the tool. In the experiments conducted we analyzed the behavior of the GA population adopting three coding schemes of solutions for the Capacitated Vehicle Routing Problem (CRVP). As a result, besides a discussion about the behavior analysis of the GA, it was found that the performance measures provided by the developed computational tool can help in the proposition and/or choice of heuristics that aim to support the refinement of the solutions generated by GA, improving its performance.

Keywords: Genetic Algorithms. Solution Encoding. Solution space. Capacitated Vehicle Routing Problem.

¹Doutor em Informática e Gestão do Conhecimento pela Universidade Nove de Junho – UNINOVE. São Paulo/SP, Brasil. stanleyjefferson@outlook.com

²Doutor em Engenharia Elétrica pela Escola Politécnica da Universidade de São Paulo – POLI/USP. Professor/pesquisador do Programa de Pós-Graduação em Informática e Gestão do Conhecimento da Universidade Nove de Junho – UNINOVE. São Paulo/SP, Brasil. saraujo@uni9.pro.br

1 Introdução

O Algoritmo Genético (AG) é um método metaheurístico de otimização baseado na teoria da evolução natural de Charles Darwin, segundo a qual os organismos de uma população que se adaptam melhor ao meio em que vivem têm mais chances de sobreviver e reproduzir, enquanto os indivíduos menos adaptados geralmente são eliminados. O processo de evolução ocorre a partir de uma população de soluções iniciais, por meio da aplicação de operadores genéticos para seleção, cruzamento e mutação dos indivíduos (HOLLAND, 1992). Cada indivíduo da população, também denominado cromossomo, é uma possível solução codificada do problema a ser resolvido pelo AG.

O AG tem sido amplamente utilizado na otimização de problemas de elevada complexidade conhecidos na literatura como *NP-Hard (Non-deterministic Polynomial Time)*, entre os quais estão o Problema de Sequenciamento da Produção (*Job Shop Scheduling Problem – JSP*), o Problema de Corte e Empacotamento (PCE) e o Problema de Roteamento de Veículos (PRV). Este último vem atraindo a atenção de muitos pesquisadores ao longo das últimas décadas em virtude da dificuldade de sua solução e da sua presença em diversas situações práticas.

Segundo Vieira (2013), o PRV consiste basicamente em determinar um roteiro (conjunto de rotas) a ser seguido por uma frota de veículos destinados a atender as demandas de um conjunto de clientes. O objetivo pode ser, por exemplo, minimizar o tempo total do percurso ou a distância total percorrida, considerando todos os veículos. Na literatura há muitas variantes do PRV, tais como Problema de Roteamento de Veículos Capacitados (PRVC), Problema de Roteamento de Veículos com Coleta e Entrega (PRVCE) e o Problema de Roteamento de Veículos com Janela de Tempo (PRVJT), entre outros. O PRVC, considerado nos experimentos conduzidos neste trabalho, é a versão mais básica e considera uma frota homogênea de veículos (em termos de capacidade) que partem e retornam para um único depósito.

Nos últimos anos o AG vem sendo amplamente empregado na solução do PRV e suas variantes devido a facilidade de sua utilização e dos bons resultados apresentados. Contudo, resolver qualquer problema de otimização usando o AG pressupõe a adoção de um esquema de codificação de soluções no cromossomo do AG e a configuração dos operadores genéticos, os quais impactam diretamente no comportamento da população do AG no espaço de soluções. Quanto maior a aderência do esquema de codificação e dos operadores genéticos ao problema, melhor será a qualidade da solução encontrada e menor será o esforço computacional para encontrá-la. Assim, entender o comportamento da população no espaço de soluções é importante para auxiliar no refinamento dos parâmetros de configuração do AG e na proposição de heurísticas que apoiem a busca por soluções de melhor qualidade, com o menor esforço computacional possível.

O espaço de soluções em problemas de otimização combinatória pode ser definido como um espaço topológico que permite que os elementos sejam comparados por meio de medidas como as distâncias de Manhattan, City Block, Euclidiana, e Hamming, entre outras (ROTHLAUF, 2011).

Nas últimas décadas tem havido muitos esforços em pesquisas visando a compreensão do espaço de soluções com objetivo de entender as dificuldades das metaheurísticas em resolver determinados problemas, bem como explorar este conhecimento a fim de desenvolver métodos mais eficientes (WATSON, 2010; LU et al., 2018). Alguns autores empregam a ideia de *Fitness Landscape*, baseada no estudo de Wright (1932), para formar uma superfície que represente o espaço de soluções com o propósito de estudar a complexidade de problemas através da análise de *Fitness Landscape*. Outros estudos como os apresentados por Boese (1995) para o Problema do Caixeiro Viajante, Merz e Freisleben (1998) para o Problema de Bipartição de Grafos e Rosa (2019) para o JSSP também contribuíram para o avanço dos estudos acerca do *Fitness Landscape* em problemas de otimização combinatória.

Apesar dos avanços, observar o comportamento de populações de algoritmos metaheurísticos como o AG no espaço de soluções não é uma tarefa fácil, principalmente porque tais espaços são de alta dimensão.

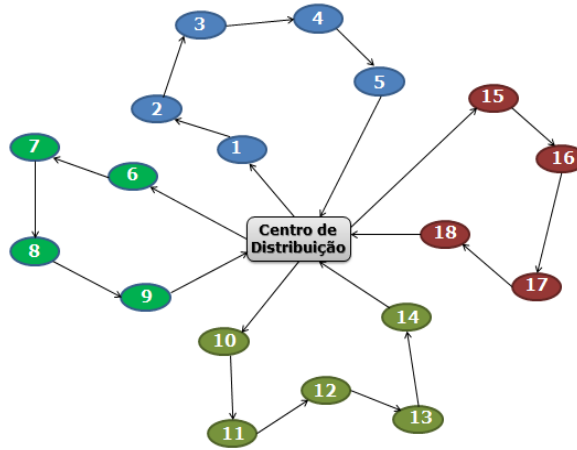
É nesse contexto que este trabalho está situado, apresentando uma ferramenta que permite investigar como os indivíduos de uma população do AG exploram o espaço de soluções em decorrência da forma como as soluções foram codificadas e a forma como os operadores genéticos foram configurados. Para avaliação da ferramenta foram conduzidos experimentos envolvendo três esquemas de codificação de soluções para o Problema de Roteamento de Veículos Capacitados (PRVC).

2 Fundamentação teórica

2.1 Problema de Roteamento de Veículos Capacitado (PRVC)

O PRVC, ilustrado na Figura 1, é a versão básica do PRV, no qual cada cliente possui demanda definida previamente, que deve ser atendida integralmente por apenas um veículo. Todos os veículos possuem capacidade idêntica, e partem de um único centro de distribuição. Nessa versão do problema apenas a restrição de capacidade do veículo é imposta, fazendo com que a soma das demandas de todos os clientes atendidos por um veículo não exceda a sua capacidade (LAPORTE, 1992).

Figura 1 – Exemplo de roteamento de veículos contendo quatro rotas definidas a partir do centro de distribuição



Fonte: Os autores.

Em outras palavras, o PRVC consiste em encontrar um conjunto de rotas, sendo que cada uma delas deve ser percorrida por um veículo, com o objetivo de minimizar o custo total do roteiro respeitando as seguintes restrições:

- cada rota deve iniciar e terminar no mesmo centro de distribuição;
- cada cliente deve ser visitado uma única vez e por um mesmo veículo;
- a soma das demandas dos clientes agrupados em uma rota não pode exceder a capacidade do veículo.

A formulação matemática para o PRVC, adaptada de Vieira (2013) para esquema de codificação binário, como o apresentado na seção 2.3.1, pode ser expressa da seguinte maneira:

$$\text{Minimizar} \quad ct = \sum_{i=0}^{nc} \sum_{\substack{j=0 \\ j \neq i}}^{nc} \sum_{k=1}^K c_{ij} x_{ijk} \quad (1)$$

$$\text{Sujeito a} \quad \sum_{k=1}^K \sum_{j=1}^{nc} x_{0jk} \leq K \quad (2)$$

$$\sum_{j=1}^{nc} x_{0jk} = \sum_{j=1}^{nc} x_{j0k} = 1, \quad k = 1, \dots, K \quad (3)$$

$$\sum_{k=1}^K \sum_{j=0}^{nc} x_{ijk} = 1, \quad i = 1, \dots, nc \quad (4)$$

$$\sum_{j=0}^{nc} x_{ijk} - \sum_{j=0}^{nc} x_{jik} = 0, \quad k = 1, \dots, K \quad i = 1, \dots, nc \quad (5)$$

$$\sum_{k=1}^K \sum_{i \in S} \sum_{j \in S} x_{ijk} \leq |S| - v(S), \forall S \subseteq V \setminus \{0\}, |S| \geq 2 \quad (6)$$

$$\sum_{i=1}^{nc} d_i \sum_{\substack{j=0 \\ j \neq i}}^{nc} x_{ijk} \leq cv, \quad k = 1, \dots, K \quad (7)$$

$$x_{ijk} \in \{0,1\}, \quad i = 1, \dots, nc, \quad j = 1, \dots, nc, \quad k = 1, \dots, K \quad (8)$$

Em que d_i : demanda do cliente i ; k : veículo; K : conjunto de veículos; S : conjunto de clientes; nc : Número de clientes; $v(S)$: número mínimo de veículos para atender S ; c_{ij} : custo do percurso do cliente i ao cliente j ; ct : custo total do roteiro; x_{ijk} : percurso do cliente i ao cliente j feito pelo veículo k .

A equação 2 assegura a utilização de K veículos partindo do centro de distribuição. A equação 3 garante que cada rota tenha seu início e término no centro de distribuição.

A equação 4 define que os clientes devem ser atendidos exatamente uma vez e, a equação 5 mantém o fluxo, garantindo que o veículo chegue e saia do cliente, evitando que a rota termine precocemente. Em adição, a equação 6 evita que sejam formadas rotas que não incluam o centro de distribuição. Nesta restrição, $v(S)$ representa o número mínimo de veículos necessários para atender o conjunto de clientes S .

Para assegurar que o número de veículos usados para atender aos clientes do conjunto S não seja inferior a $v(S)$, a restrição 6 estabelece indiretamente que a capacidade do veículo não seja excedida. Contudo, para deixar explícita a formulação da restrição de capacidade, apresenta-se a equação 7.

Por fim, a equação 9 é usada para avaliar as soluções. Tal equação reflete o valor da função objetivo e envolve o número de veículos empregados na solução, restrições violadas (equações 2 a 7) e o custo total das rotas (equação 1).

$$FO = (KW_v) + (nrW_r) + ct \quad (9)$$

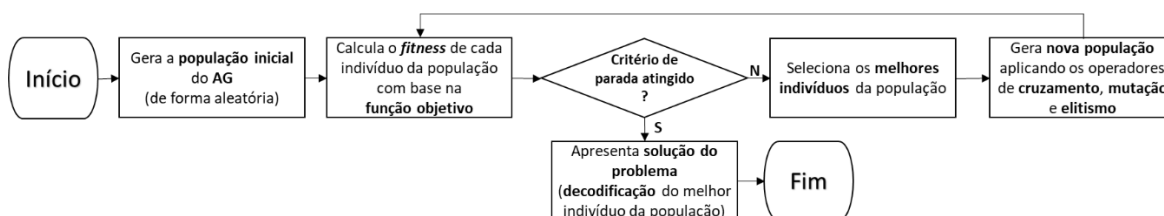
Em que: W_v é o peso atribuído ao número de veículos utilizados na solução; nr é o número de restrições violadas e W_r é o peso atribuído às restrições violadas.

2.2 Algoritmo Genético (AG)

O AG é um método de busca e otimização inspirado no processo de seleção natural (GOLDBERG, 1989). De modo geral, o funcionamento do AG consiste em manter uma população de indivíduos que representam possíveis soluções para determinado problema e que evoluem ao longo das gerações (iteração) através de um processo de competição, no qual os mais aptos (melhores soluções) têm maiores chances de sobreviver e de se reproduzir. Um indivíduo do AG é representado por um cromossomo (normalmente uma *string* binária), composto por genes (*substrings*) e alelos (bits).

O funcionamento do AG pode ser descrito, de forma resumida, da seguinte maneira: inicia-se uma população contendo um número pré-definido de indivíduos, cada indivíduo representando uma possível solução do problema. Para cada indivíduo se atribui um coeficiente de aptidão ou *fitness*, geralmente representado pelo valor da função objetivo, que indica quão apto é o indivíduo (qualidade da solução) em relação à população. Assim, os indivíduos são selecionados com base nas suas aptidões e passam suas características para a próxima geração, por meio da aplicação dos operadores genéticos de cruzamento, mutação e elitismo, com o intuito de prover uma nova geração formada por indivíduos mais aptos. Este procedimento iterativo continua até que um critério de parada seja alcançado (por exemplo o número de gerações), como ilustrado na Figura 2.

Figura 2 – Esquema de funcionamento do AG



Fonte: Os autores.

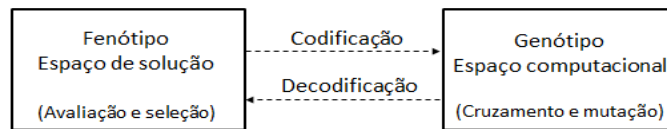
Alguns outros conceitos importantes associados ao AG, relacionados às operações de codificação e decodificação da solução, são:

- Genótipo: está relacionado à população no espaço computacional, no qual as soluções são representadas para serem facilmente entendidas e manipuladas computacionalmente (BROOKER, 2012);
- Fenótipo: está relacionado com a população no espaço de soluções do mundo real, em que as soluções são representadas para serem interpretadas em situações do mundo real (BROOKER,

2012; KUMAR, 2013), ou seja, é a estrutura construída a partir das informações do genótipo que permite a decodificação do cromossomo;

- Codificação e Decodificação: para Kumar (2013) codificação é uma operação que transforma uma solução do espaço fenótipo para o espaço genotípico, enquanto a decodificação é responsável pela transformação inversa, como mostra a Figura 3.

Figura 3 – Operação de codificação e decodificação



Fonte: Adaptado de Kumar (2013).

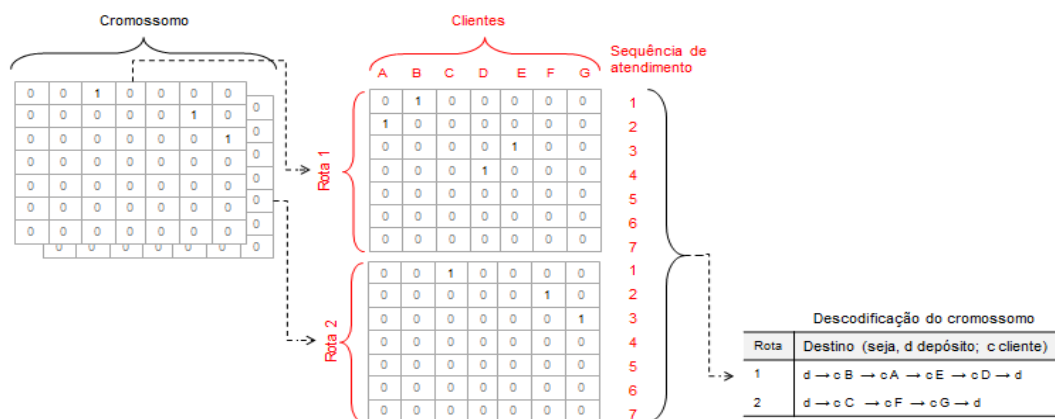
Com base no exposto nesta seção, percebe-se a importância do esquema de codificação de soluções para o bom desempenho do AG.

2.3 Esquemas de codificação de soluções do PRVC considerados neste trabalho

2.3.1 Esquema de codificação A

O esquema de codificação de solução do PRVC ilustrado na Figura 4 foi utilizado nos trabalhos de Vieira (2013) e Lima et al. (2015). Ele emprega uma matriz binária tridimensional de M colunas, N linhas e P profundidade, onde M representa o número de clientes a serem atendidos, N representa a sequência de atendimento ao cliente e P é definido pelo número de veículos (rota) necessários para atender a demanda total. A posição da coluna que recebe valor 1 no alelo indica o cliente a ser visitado enquanto a posição da linha indica ordem de atendimento do cliente. Já a profundidade representa o veículo atribuído para atender um dado cliente.

Figura 4 – Esquema de codificação/decodificação empregado em Lima et al. (2015)

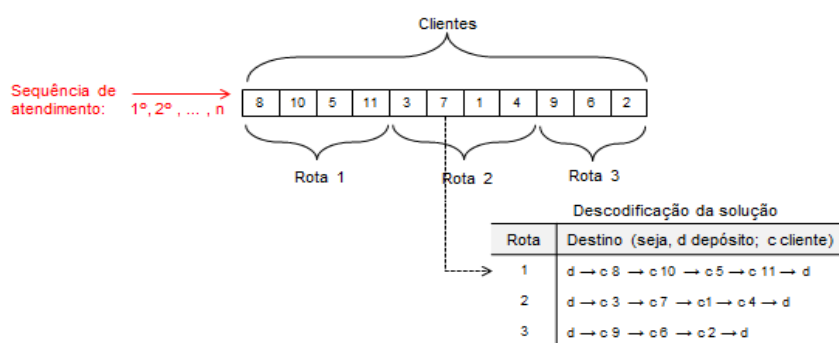


Fonte: Lima et al. (2015).

2.3.2 Esquema de codificação B

Lee e Nazif (2012) empregaram um vetor de números inteiros de comprimento N , onde N representa o número de clientes a serem atendidos e, cada gene do vetor é um número que corresponde a um determinado cliente, como ilustrado na Figura 5. A sequência dos genes no vetor determina a ordem de atendimento dos clientes e, o conjunto de clientes que compõem cada rota, são delimitados pela capacidade dos veículos, de modo que os clientes são atribuídos a um determinado veículo na sequência em que aparece no vetor. A medida que a capacidade do veículo é excedida uma nova rota é iniciada.

Figura 5 – Esquema de codificação/decodificação proposto por Lee e Nazif (2011)

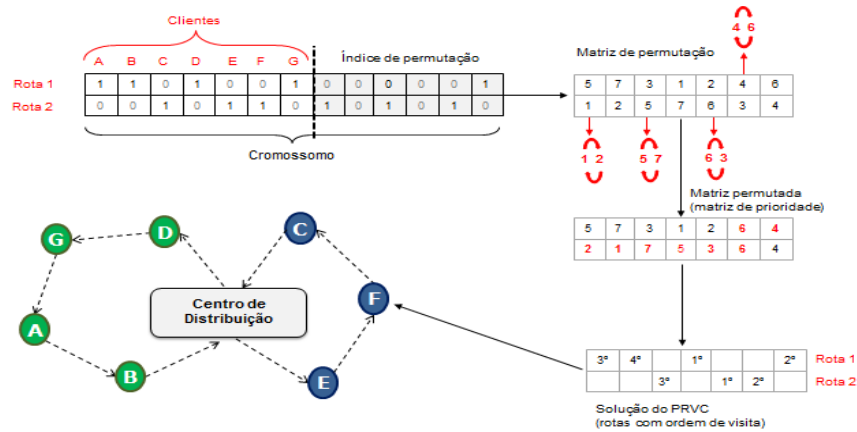


Fonte: Adaptado de Lee e Nazif (2011).

2.3.3 Esquema de codificação C

Lima e Araújo (2018) apresentaram um esquema ilustrado na Figura 6 para codificação de soluções do PRVC. O esquema consiste em uma matriz binária de M colunas por N linhas, sendo M o resultado da equação $nc * 2 - 1$, onde nc representa o número de clientes. Para a otimização do roteiro parte do cromossomo, denominada índice de permutação, é usada para permutar os clientes associados aos veículos.

Figura 6 – Esquema de codificação/decodificação proposto por Lima e Araújo (2018)



Fonte: Adaptado de Lima e Araújo (2018).

3 Metodologia

Para avaliação da ferramenta, os esquemas de codificação apresentados na seção 2.3 (A, B e C) foram aplicados em 3 instâncias (cenários do PRVC) do conjunto proposto por Christofides (1985), quais sejam: E-n13-k4, E-n33-k4 e E-n76-k14. Elas foram classificadas pelo tamanho, levando em conta a distribuição do número de clientes (n) e veículos (k) indicados em cada instância do conjunto. Por exemplo, a instância E-n13-k4 considera um cenário do PRVC com 13 clientes e 4 veículos. Assim, E-n13-k4 foi classificada como pequena (poucos veículos e poucos clientes), E-n33-k4 como média (número médio de veículos e/ou clientes) e E-n76-k14 como grande (muitos veículos e/ou clientes). As soluções ótimas para essas instâncias foram extraídas do trabalho de Reinelt e Wenger (2006).

Os parâmetros empregados no AG, incluindo a configuração dos operadores genéticos, foram definidos empiricamente e estão listados na Tabela 1.

Tabela 1 – Parâmetros de configuração do AG

Parâmetro	Valor adotado
Tamanho da População (número de indivíduos)	200
Número de Gerações	50
Número de indivíduos factíveis injetados na população inicial	2
Taxa de cruzamento	0,8
Pontos de corte	1
Taxa de mutação	0,1
Taxa de elitismo	0,05
Método de Seleção	Roleta
Critério de parada	Número de gerações

Fonte: Os autores.

Com o objetivo de realizar uma comparação mais precisa, para os três esquemas de codificação considerou-se a mesma população inicial gerada de maneira aleatória, com distribuição uniforme. Em adição, foram inseridos na população inicial dois indivíduos gerados pela heurística de *Gillett & Miller* (Gillett e Miller, 1974), fazendo com que o AG já iniciasse com soluções factíveis.

Por fim, a análise do comportamento do AG em função de cada esquema de codificação foi feita com base em três das medidas fornecidas pela ferramenta apresentada neste trabalho: convergência, percentual de soluções não factíveis e diversidade.

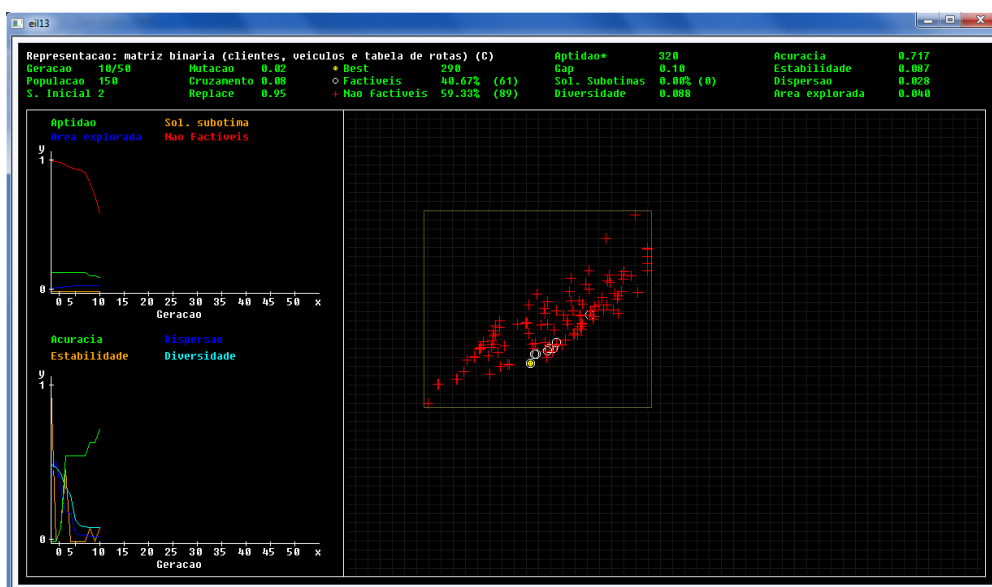
4 Resultados e discussão

4.1 Ferramenta computacional proposta

A ferramenta computacional desenvolvida permite a visualização e análise dos indivíduos da população do AG, ou ainda de outra metaheurística baseada em população, em um espaço bidimensional, como ilustrado na Figura 7. Em adição, os resultados gerados pela ferramenta são gravados em arquivo texto formato CSV (*Comma Separated Values*), que pode ser lido no Microsoft Excel, e em vídeo no formato AVI (*Audio Video Interleave*), que propicia uma análise mais detalhada do comportamento das populações.

Para o desenvolvimento da ferramenta empregou-se a linguagem de programação C/C++ e as bibliotecas *Proeikon*¹, para a manipulação e processamento de imagens, e *GALib*² para manipulação dos componentes típicos do AG.

Figura 7 – Interface da ferramenta computacional desenvolvida



Fonte: Os autores.

¹ <http://www.lps.usp.br/hae/software/proeikon.html>

² <http://lancet.mit.edu/ga/>

Cada indivíduo da população é projetado de um espaço de soluções n-dimensional para o espaço bidimensional representado pelo painel à direita da tela mostrada na Figura 7, no qual o eixo x é representado pela distância Euclidiana e o eixo y pela distância de Hamming. Para tanto, considera-se que um indivíduo α com todos os alelos nulos é representado pelo ponto (x_0, y_0) e então cada indivíduo i de uma população p é projetado para um ponto (x_i, y_i) , definido com base no cálculo das duas distâncias entre os indivíduos i e α . Vale ressaltar que essas distâncias são amplamente empregadas na literatura para medir a similaridade entre indivíduos de populações geradas por algoritmos metaheurísticos (LIMA; ARAÚJO, 2018).

Outra opção para visualização do espaço de soluções seria adotar métodos específicos para projeção de espaços n-dimensionais para espaços bi ou tri-dimensionais como, por exemplo, o *Principal Component Analysis* (PCA), *T-Distributed Stochastic Neighbor Embedding* (t-SNE) e o *Least Square Projection* (LSP). Entretanto, tais métodos consideram todo o conjunto de dados (população) para determinar a posição de um vetor (indivíduo) no espaço de dimensão reduzida. Desta forma, um mesmo indivíduo pode ser projetado em posições diferentes ao longo das gerações caso haja mudança em qualquer outro indivíduo pertencente à sua população. Deste modo, para assegurar que a posição de cada indivíduo no espaço de baixa dimensão não seja alterada ao longo das gerações adotou-se o cálculo das distâncias supracitadas, permitindo assim que o comportamento dos indivíduos da população no espaço de soluções seja analisado de maneira dinâmica.

O painel esquerdo exibe dois gráficos plotados em tempo de execução. O primeiro apresenta a medida de aptidão ou *fitness* (função de avaliação que atribui um valor a cada indivíduo da população), a área explorada (percentual do espaço de soluções explorado) e os percentuais de soluções factíveis e não factíveis. Já o segundo gráfico ilustra a acurácia (visa determinar a localização da melhor solução encontrada, em um intervalo definido por um limite inferior e superior), a estabilidade (reflete o quanto um algoritmo é capaz de se estabilizar após uma mudança no cenário), a dispersão média (define o quão distantes estão os indivíduos, em relação às suas posições no espaço de soluções) e a diversidade da população (variação genética dos indivíduos da população do AG em cada geração).

A ferramenta também apresenta uma barra de status, localizada na parte superior, a qual exibe os parâmetros empregados no AG, tais como: esquema de codificação, tamanho da população, número de iterações (gerações), iteração atual, número de indivíduos factíveis inseridos na população inicial, taxa de cruzamento, taxa de mutação, elitismo e o valor da função objetivo da melhor solução encontrada literatura para a instância que está sendo resolvida, simbolizado por FO_{best} neste trabalho.

Ainda na barra de status são mostrados o custo computacional (tempo em segundos entre o início e o término da execução do AG na otimização de uma instância) e os valores calculados ao longo da execução do AG para 8 medidas distintas, listadas a seguir, sendo as 4 primeiras extraídas da literatura e as 4 últimas propostas neste trabalho.

- Acurácia: visa determinar a localização da melhor solução encontrada, dentro do intervalo definido por um limite inferior representando pela pior solução encontrada e um limite superior, indicado pela melhor solução conhecida (WEICKER, 2002).
- Diversidade da População: esta medida expressa, por meio de uma taxa, a diversidade da população. O valor 0 para esta taxa indica que todos os indivíduos da população são iguais, enquanto o valor 1 sinaliza que todos os indivíduos são completamente diferentes do (LINDEN, 2012).
- Estabilidade: indica o quanto o algoritmo é capaz de manter sua estabilidade. Um algoritmo estável mantém uma acurácia por inúmeras iterações ou gerações (BEN-ROMDHANE et al., 2013).
- GAP: expressa a diferença entre a melhor solução encontrada pelo algoritmo que está sendo executado para solucionar um problema (FO) e a melhor solução conhecida na literatura para esse problema (FO_{best}). Seu cálculo é feito da seguinte maneira:
$$GAP = (FO - FO_{best})/FO_{best}.$$
- Área Explorada: esta medida expressa, por meio de uma taxa que varia de 0 a 1, o percentual da área do espaço de soluções explorada durante a execução do AG. Quanto maior a taxa, maior foi a área explorada, indicando que o algoritmo teve maior probabilidade de explorar pontos mais promissores do espaço de soluções.
- Dispersão Média: descreve o quão afastados (dispersos) estão os indivíduos da população, ou seja, quanto maior a dispersão média maior será a área explorada no espaço de soluções.
- Percentual de Indivíduos Factíveis: expresso por meio de uma taxa que reflete o percentual de soluções factíveis presentes na população, mostrando assim a capacidade do algoritmo de converter soluções não factíveis em factíveis. Dessa forma, o Percentual de Indivíduos Não Factíveis é dado por: 1- Percentual de Indivíduos Factíveis.
- Percentual de soluções subótimas: essa medida revela, por meio de uma taxa, o percentual de soluções subótimas na população. Neste trabalho considera-se como soluções subótimas aquelas com $GAP \leq 0,10$.

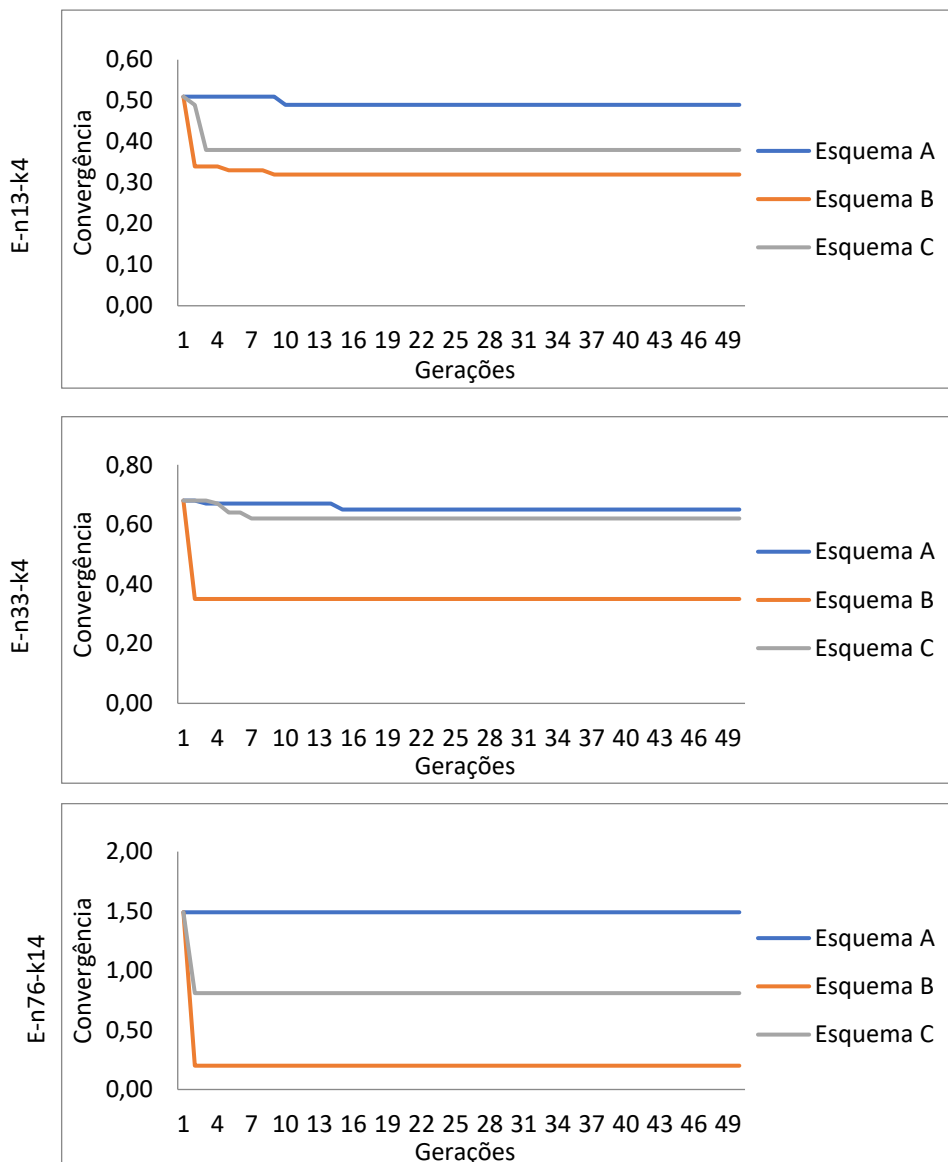
4.2 Análise comparativa do comportamento da população do AG em função do esquema de codificação adotado

Nesta seção são apresentados os resultados obtidos nos experimentos acerca da análise do comportamento do AG em função dos três esquemas de codificação de solução do PRVC apresentados na seção 2.3, sendo os esquemas A e C binários e o esquema B inteiro. Embora a ferramenta computacional forneça inúmeras medidas de desempenho, algumas delas descrevem o

comportamento do AG de forma semelhante. Assim optou-se por considerar apenas as seguintes medidas: convergência, percentual de soluções não factíveis e diversidade.

Em relação a convergência, observa-se nos gráficos mostrados na Figura 8 que o esquema de codificação A apresentou uma sutil melhora na convergência para as instâncias E-n13-k4 e E-n33-k4, entretanto na instância E-n76-k14 esse esquema não conseguiu convergir para pontos mais promissores no espaço de soluções, não conseguindo assim melhorar a solução inicial. Isso pode ser explicado pelo fato de ser um esquema de codificação esparsa aplicado a uma instância relativamente grande, o que pode sugerir a necessidade de um número maior de gerações.

Figura 8 - Gráficos comparativos de convergência



Fonte: Os autores.

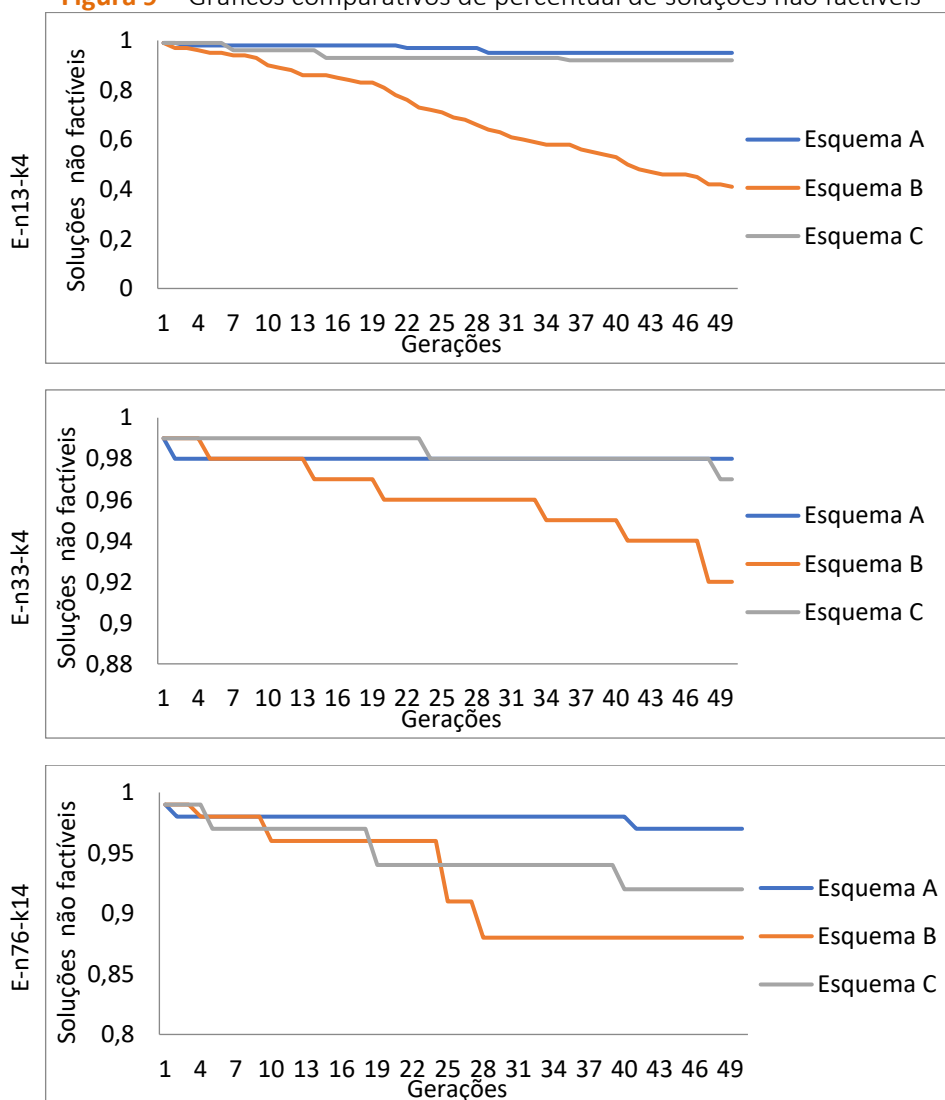
Por outro lado, o esquema de codificação B apresentou a melhor convergência para as três instâncias consideradas, além de apresentar uma convergência rápida para pontos de ótimos locais, estagnando na 10ª geração no caso da instância E-n13-k4 e na 3ª geração nas demais instâncias.

O esquema de codificação C apresentou um comportamento de evolução parecido ao do esquema B, ou seja, uma convergência prematura (nas primeiras gerações), em seguida uma estagnação em pontos de ótimos locais.

No que tange o percentual de soluções não factíveis, observa-se nos gráficos apresentados na Figura 9 que o esquema de codificação A não apresentou uma redução significativa desse percentual, alcançando para as instâncias E-n13-k4, E-n33-k4 e E-n76-k14, respectivamente, 92%, 98% e 97%. Em adição, este esquema apresenta pouca melhora entre uma geração e a seguinte. Assim, o esquema de codificação A se mostra suscetível a gerar soluções não factíveis.

O esquema de codificação B apresentou os menores percentuais de soluções não factíveis: 42%, 92% e 87%. Observa-se também, sobretudo nas instâncias E-n13-k4 e E-n33-k4, uma tendência de queda neste indicador, ou seja, este índice diminui a cada geração, apontando que com um número maior de gerações o algoritmo conseguiria reduzir ainda mais estes percentuais. Em adição, esses resultados apontam que esquemas de codificação usando vetores de inteiros apresentam bom desempenho no aspecto de geração de soluções factíveis.

Figura 9 – Gráficos comparativos de percentual de soluções não factíveis



Fonte: Os autores.

Os gráficos da Figura 9 também evidenciam que o esquema de codificação C apresentou pouca redução neste indicador, alcançando para as instâncias E-n13-k4, E-n33-k4 e E-n76-k14 os percentuais de 93%, 97% e 92%, respectivamente. Vale ressaltar que os esquemas A e C são codificações binárias, que podem ser menos eficientes no que tange a transformação de soluções não factíveis em soluções factíveis.

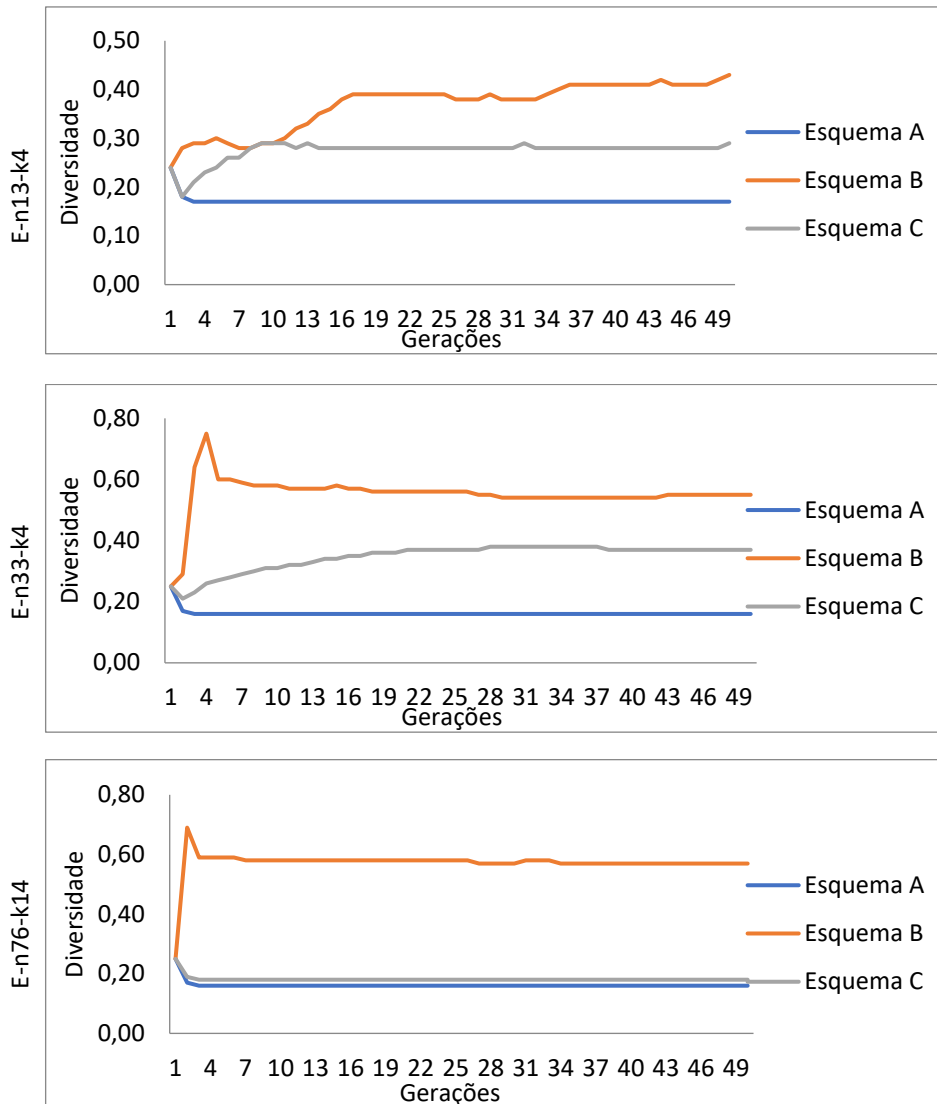
A diversidade é outro aspecto importante no AG, pois ela previne a sua estagnação prematura, ajudando a evitar que ele fique “preso” em ótimos locais. Os gráficos da Figura 10 evidenciam que o esquema de codificação A apresenta uma tendência de perder rapidamente a diversidade da população, sendo que, em média, na 3ª geração esta medida já se encontra baixa, apresentando para as instâncias E-n13-k4, E-n33-k4 e E-n76-k14 os seguintes percentuais de 17%, 16% e 16%, respectivamente.

O esquema de codificação B apresentou boa diversidade, chegando a alcançar, em determinadas gerações, os seguintes percentuais: 42%, 75% e 69% e, considerando as três instâncias, esse esquema apresentou, em média, um percentual acima de 50% evidenciando que ele possibilita uma boa diversidade de indivíduos na população do AG.

O esquema de codificação C apresentou uma diversidade menor do que esquema B e maior do que o esquema A para três instâncias, alcançando 28%, 37% e 18%, respectivamente. Esses resultados demonstram que neste aspecto o esquema de codificação B apresenta o melhor desempenho em comparação com os esquemas A e C.

Fazendo uma análise geral dos resultados obtidos nos experimentos, pode-se evidenciar que o esquema de codificação de solução adotado tem forte impacto sobre diversos aspectos no comportamento da população, bem como no desempenho do AG. Pode-se constatar também que os esquemas de codificação A permite uma exploração intensa em sub-regiões do espaço de soluções. Por outro lado, o esquema de codificação B apresentou bom desempenho no que tange a convergência, diversidade e redução do percentual de soluções não factíveis. Já o esquema de codificação C apresentou desempenho apenas satisfatório para esses mesmos indicadores.

Figura 10 – Gráficos comparativos de diversidade



Fonte: Os autores.

Dessa forma, caso se empregue o esquema de codificação A, recomenda-se inserir um número maior de soluções factíveis na população inicial geradas por alguma heurística gulosa, já que isso fará o AG gerar um número maior de descendentes factíveis. No caso do emprego dos esquemas de codificação B ou C, sugere-se o uso de heurísticas que auxiliem uma exploração mais profunda em sub-regiões, fazendo com que o AG explore melhor uma vizinhança (busca local) antes de “saltar” para outras regiões do espaço de soluções.

5 Conclusões

Os experimentos realizados evidenciaram que a ferramenta proposta é uma boa alternativa para avaliar o comportamento dos indivíduos da população do AG ao longo das gerações, principalmente em virtude dos dados numéricos produzidos e gravados, que podem ser diretamente analisados no Excel. Não obstante, os vídeos gravados durante as execuções do AG podem ser

reproduzidos inúmeras vezes, por meio de qualquer aplicativo que leia o formato .AVI, permitindo uma análise mais detalhada do espaço de soluções. Em adição, pode-se constatar que as medidas de desempenho implementadas na ferramenta podem compor uma heurística com objetivo de apoiar o processo de refinamento do AG, melhorando assim o seu desempenho. Por fim, ressalta-se que a ferramenta pode trabalhar em conjunto com outros algoritmos metaheurísticos baseados em população diferentes do AG.

Sobre os resultados dos experimentos envolvendo as instâncias do PRVC, constatou-se que o esquema de codificação usando matriz binária é menos eficiente no que tange a conversão de soluções não factíveis em factíveis, além de produzir baixa diversidade populacional dificultando a convergência do AG. Já o esquema de codificação usando vetores de números inteiros se mostrou mais eficiente em gerar soluções factíveis, bem como em proporcionar boa diversidade da população, auxiliando assim o processo de convergência do AG. Talvez seja por este motivo que este tipo de codificação vem sendo empregado nos trabalhos atuais, embora essa explicação não seja fornecida em tais trabalhos.

Em trabalhos futuros pretende-se realizar um estudo mais amplo do comportamento do AG na solução do PRVC levando em conta todas as medidas fornecidas pela ferramenta e diferentes configurações dos operadores genéticos, implementar outras técnicas de projeção multidimensional de dados como por exemplo o PCA e o LSP, e avaliar o uso da ferramenta com o uso de outros algoritmos metaheurísticos e/ou outros problemas de otimização.

Referências

- Andrade, M. M. (2001). Como Preparar Trabalhos para Cursos de Pós Graduação. 4. ed. São Paulo: Atlas.
- Ben-Romdhane, H., Alba, E., & Krichen, S. (2013). Best practices in measuring algorithm performance for dynamic optimization problems. *Soft Computing*, 17(6), 1005-1017. <https://doi.org/10.1007/s00500-013-0989-7>
- Boese, K. D., Kahng, A. B., & Muddu, S. (1994). A new adaptive multi-start technique for combinatorial global optimizations. *Operations Research Letters*, 16(2), 101-113. [https://doi.org/10.1016/0167-6377\(94\)90065-5](https://doi.org/10.1016/0167-6377(94)90065-5)
- Brooker, R. J. (2012) Concepts of genetics. New York: McGraw-Hill.
- Christofides, N. (1985). Vehicle Routing. In: The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization, Lawer, E.L., Lenstra, J.K., kan, A.H.G.R., Shmoys, D.B (eds), Great Britain.: John Wiley & Sons Ltd,
- Gillett, B. E., & Miller, L. R. (1974). A heuristic algorithm for the vehicle-dispatch problem. *Operations research*, 22(2), 340-349. <https://doi.org/10.1287/opre.22.2.340>
- Goldberg, D. E. (1989) Genetic Algorithms in search, optimization and machine learning. U.S.A., Addison-Wesley Publishing Company,

- Holland, J. H. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.
- Kumar, A. (2013). Encoding schemes in genetic algorithm. *International Journal of Advanced Research in IT and Engineering*, 2(3), 1-7.
- Laporte, G. (1992). The vehicle routing problem: An overview of exact and approximate algorithms. *European journal of operational research*, 59(3), 345-358.
[https://doi.org/10.1016/0377-2217\(92\)90192-C](https://doi.org/10.1016/0377-2217(92)90192-C)
- Lima S.J.A., & Araújo S.A. (2018) A New Binary Encoding Scheme in Genetic Algorithm for Solving the Capacitated Vehicle Routing Problem. In: Bioinspired Optimization Methods and Their Applications. BIOMA 2018. Korošec P., Melab N., Talbi EG. (eds) *Lecture Notes in Computer Science*, 10835, 174-184. https://doi.org/10.1007/978-3-319-91641-5_15.
- Lima, S. J. A., SANTOS, R., & ARAÚJO, S. (2015). Otimização do problema de roteamento de veículos capacitado usando Algoritmos Genéticos e as heurísticas de Gillet e Miller e descida de encosta. *Anais do XXXV Encontro Nacional de Engenharia de Produção*. Rio de Janeiro: ABEPRO, 1, pp. 1-15.
- Linden, R. (2012). Algoritmos Genéticos. *Editora Ciência Moderna*, 3. ed.
- Lu, H., Shi, J., Fei, Z., Zhou, Q., & Mao, K. (2018). Analysis of the similarities and differences of job-based scheduling problems. *European Journal of Operational Research*, 270(3), 809-825.
<https://doi.org/10.1016/j.ejor.2018.01.051>
- Merz, P., & Freisleben, B. (1998, September). Memetic algorithms and the fitness landscape of the graph bi-partitioning problem. In: *International Conference on Parallel Problem Solving from Nature*, pp. 765-774, Springer, Berlin, Heidelberg.
- Nazif, H., & Lee, L. S. (2012). Optimised crossover genetic algorithm for capacitated vehicle routing problem. *Applied Mathematical Modelling*, 36(5), 2110-2117.
<https://doi.org/10.1016/j.apm.2011.08.010>
- Rosa, A. F. C. (2019). *Algoritmo Genético híbrido baseado na análise de componentes principais co Fitness Landscape para O Job Shop Scheduling Problem*. São Paulo: UNINOVE, 2019. 91p. Tese (Doutorado em Informática e Gestão do Conhecimento) – Programa de Pós-graduação em Informática e Gestão do Conhecimento, Universidade Nove de Julho, São Paulo.
- Rothlauf, F. (2011). *Design of modern heuristics: principles and application*. Springer Science & Business Media.
- Vieira, H. P. (2013). Metaheurística para a solução de problemas de roteamento de veículos com janela de tempo. Campinas: UNICAMP, 2013. 108p. Dissertação (Mestrado em Matemática Aplicada) – Instituto de Matemática, Estatística e Computação Científica, Universidade Estadual de Campinas, Campinas.
- Watson, J. P. (2010). An introduction to fitness landscape analysis and cost models for local search. In: *Handbook of metaheuristics*, pp. 599-623. Springer, Boston, MA.
- Weicker, K. (2002, September). Performance measures for dynamic environments. In: *International Conference on Parallel Problem Solving from Nature*, pp. 64-73. Springer, Berlin, Heidelberg.

Recebido em: 01 nov. 2019 / **Aprovado em:** 12 dez. 2019

Para referenciar este texto

American Psychological Association (APA)

Lima, S. J. d A., & Araújo, S. A. de (2020, out./dez.). Ferramenta computacional para análise do comportamento da população do algoritmo genético no espaço de soluções. *Exacta*, 18(4), 725-743. <https://doi.org/10.5585/exactaep.v18n4.15959>.