

# COMPUTAÇÃO NATURAL PARA SOLUÇÃO DE PROBLEMAS MULTICRITÉRIOS E MULTIOBJETIVOS COM USO DE VARIÁVEIS FUZZY

*Data de submissão: 28/04/2023*

*Data de aceite: 02/06/2023*

### **José Guilherme Picolo**

Universidade São Francisco  
Itatiba – São Paulo  
<http://lattes.cnpq.br/1457140177639434>

### **Rafael Prazeres da Silva**

Universidade São Francisco  
Itatiba – São Paulo  
<http://lattes.cnpq.br/7143350735711256>

### **Silvio Petrolino Neto**

Universidade São Francisco  
Itatiba – São Paulo  
<http://lattes.cnpq.br/9100171823896988>

**RESUMO.** Na área da computação é recorrente o uso de algoritmos heurísticos na busca de soluções otimizadas para problemas de diversas áreas do conhecimento. O presente artigo tem como objetivo evidenciar que o uso das técnicas de algoritmos genéticos e sistemas imunológicos artificiais aliados à técnica de representação de variáveis por meio de lógica fuzzy são adequadas para a resolução de problemas multicritérios e multiobjetivos. Para isso, foram implementadas as técnicas descritas para a resolução do problema de alocação de benefícios para o quadro de colaboradores

de uma empresa, visando encontrar a maior satisfação dos funcionários e o menor custo para a empresa. Ao final, após analisar os resultados obtidos, foi possível determinar que tanto o algoritmo genético como o sistema imunológico artificial se mostraram ferramentas adequadas na resolução do problema proposto, atingindo soluções bem próximas ao ideal.

**PALAVRAS-CHAVE:** algoritmos genéticos, sistema imunológico artificial, lógica difusa, problemas multicritérios, problemas multiobjetivos.

### NATURAL COMPUTING FOR SOLUTION OF MULTI-CRITERIA AND MULTI-OBJECTIVE PROBLEMS USING FUZZY VARIABLES

**ABSTRACT.** In computing area is recurrent the use of heuristic algorithms in the search of solutions optimized for problems of several areas of knowledge. The article aims to show the use of genetic algorithm and artificial immune systems techniques align the technique of representing variables by fuzzy logic are adequate for the resolution of multi-criteria and multi-objective problems. For this purpose, the techniques were implemented to solve the benefit allocation problem to the employees of a company,

aiming to find the highest satisfaction from employees and the lower cost for the company. In the end, after analyzing the results, it was possible to determine that both techniques of genetic algorithm and artificial immune systems proved to be adequate tools in the solution of the proposed problem, reaching solutions very close to the ideal.

**KEYWORDS:** genetic algorithm, fuzzy logic, multi-criteria optimization, multi-objective optimization

## 1 | INTRODUÇÃO

A computação natural descreve todo sistema na área de computação implementado com inspiração ou utilização de algum mecanismo natural ou biológico de processamento de informação para o desenvolvimento de sistemas artificiais (GOEDERT; PAULA FILHO; BLANCO, 2017). A área da computação natural pode ser dividida em três grandes áreas: Computação inspirada na natureza, Estudos sobre a natureza através da computação e Computação com mecanismos naturais. O primeiro ramo da computação natural é o ramo mais antigo e consolidado e compreende, dentre vários modelos, os algoritmos genéticos, redes neurais artificiais, sistemas imunológicos artificiais, entre outros. O trabalho pioneiro nesta área surgiu na década de quarenta (CASTRO et al., 2019).

Este trabalho de pesquisa propõe mostrar que os algoritmos de computação natural da área da computação inspirada na natureza são uma ferramenta adequada para encontrar a solução de problemas multicritérios e multiobjetivos. Na atualidade, os processos de tomada de decisão no âmbito empresarial possuem um alto nível de complexidade visto que quando se trata de um problema de otimização, a complexidade se dá pelo fato do grande volume de informações disponíveis, além dos diversos objetivos que devem ser atendidos de forma simultânea na busca de uma solução ótima. Neste contexto, surge a análise de multicritérios, que consiste em uma ferramenta que tem como propósito comparar diferentes soluções, baseado nos diversos critérios, com a intenção de direcionar os tomadores de decisão para uma escolha assertiva (GOMES; COSTA, 2015) e a análise multiobjetivo que se caracteriza por possuir duas ou mais funções de objetivo para serem otimizados simultaneamente, sendo que na maioria dos casos tais funções podem ser conflitantes (COELLO, 2006). Visando maior assertividade e precisão no resultado obtido, os algoritmos desenvolvidos serão integrados à lógica *fuzzy*.

A lógica *fuzzy*, ou lógica difusa, é apresentado pelos autores GOMIDE, GUDWIN E TANSCHKEIT (2019) como uma lógica aproximada, no qual uma proposição assume um valor de um elemento de um conjunto, que difere da lógica tradicional que somente aceita dois valores: verdadeiro ou falso e, por isso, essa técnica é muito poderosa para manusear informações qualitativas. Diante do apresentado, esse projeto de pesquisa também objetiva mostrar que o uso de lógica *fuzzy* para a definição dos critérios permitirá resultados mais precisos na resolução de problemas multicritérios e multiobjetivos.

O cenário que esse projeto de pesquisa se propõe permeia o âmbito empresarial e,

o problema a ser solucionado, consiste na otimização da alocação de benefícios de uma empresa dentre o quadro de seus funcionários, visando a maior satisfação dos mesmos e o menor investimento por parte da empresa. Muitas empresas possuem dificuldade em alocar os seus benefícios, pois não possuem uma ferramenta adequada para um problema que pode se tornar muito complexo, podendo resultar em gastos excessivos e/ou baixa satisfação dos funcionários. A criação de um algoritmo para analisar as informações de funcionários de uma base de dados e que, ao final, evidenciará a melhor solução trará para a empresa diretamente uma economia mensal com os gastos em benefícios e maior satisfação geral dos funcionários.

## 1.1 Algoritmos genéticos

No âmbito de desenvolvimento de algoritmos, visando a área de sistemas inteligentes, tem-se como um dos principais solucionadores de problemas a técnica de algoritmos genéticos. Através do nome desta técnica é possível identificar a relação com a biologia, pois ela utiliza os conceitos abordados na teoria de Charles Darwin (1859-1993), conhecida como a Teoria da Evolução. Os autores SILVA e PIGNATA (2019) expõem que essa teoria afirma que é o ambiente, através da seleção natural, que determina a relevância de uma característica para um indivíduo e, dentre esses indivíduos, os mais bem adaptados possuem maiores chances de sobrevivência, deixando um maior número de descendentes.

Utilizando os conceitos de Darwin para iniciar o desenvolvimento de um algoritmo genético, o autor PETROLI NETO (2011) afirma que é necessária a definição de duas condições: a codificação do cromossomo e uma função de avaliação. Ao codificar um cromossomo, tem-se como objetivo representar a solução do problema; é importante que essa codificação possua genes, pois durante a reprodução são os genes que são herdados pelos descendentes. Ao definir a função de avaliação, tem-se o propósito de definir um meio de avaliar um indivíduo dentro de uma população, pois, assim, os indivíduos que possuem as melhores características, são mais propensos a gerar descendentes, enquanto os que possuem as piores, são descartados.

A reprodução é o processo essencial em um algoritmo genético, pois produz gerações de indivíduos que tendem a ser melhores que os indivíduos da geração anterior. O autor CARVALHO (2009) define reprodução como o processo em que indivíduos são selecionados para gerar descendentes que contém material genético herdado através de recombinação genética (crossover) e que pode ou não sofrer mutações; esse processo é repetido até que uma solução satisfatória seja atingida. MOTA (2007) define alguns termos utilizados; segundo o autor, seleção pode ser compreendida como a eleição de quais indivíduos da população atual serão utilizados para gerar descendentes; recombinação pode ser entendido como o processo onde os genes dos indivíduos selecionados serão combinados e, por fim, a mutação, que pode ser compreendida como o processo de alteração dos genes de um cromossomo de um indivíduo. A Figura 1 apresenta um

algoritmo padrão de um algoritmo genético; é possível observar que o algoritmo recebe dois parâmetros  $pc$  e  $pm$ , os quais estão relacionados com a taxa de recombinação e taxa de mutação, respectivamente.

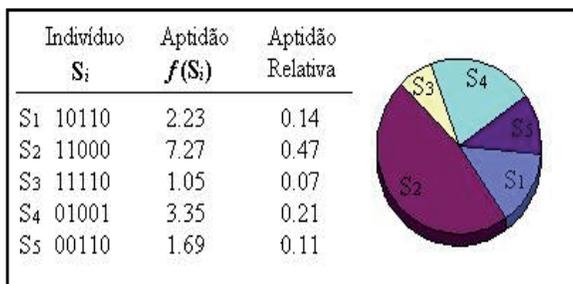
```

procedimento [P] = padrão_AE ( $pc$ ,  $pm$ )
  inicialização P
   $f \leftarrow$  avaliação (P)
  P  $\leftarrow$  seleção (P,  $f$ )
   $t \leftarrow 1$ 
  enquanto não_criterio_parada faça,
    P  $\leftarrow$  reprodução (P,  $f$ ,  $pc$ )
    P  $\leftarrow$  variação (P,  $pm$ )
     $f \leftarrow$  avaliação (P)
    P  $\leftarrow$  seleção (P,  $f$ )
     $t \leftarrow t + 1$ 
  fim enquanto
fim procedimento

```

**Figura 1** - Algoritmo padrão de um Algoritmo Genético. Adaptado de CASTRO (2006).

Nesse projeto, para selecionar dois indivíduos para a etapa de cruzamento, será utilizado o Método da Roleta. Como o autor ZAMBIASI (2011) descreve, esse método tem como fundamento separar uma parte maior de uma roleta para os indivíduos mais aptos e, aos que possuem pouca aptidão, são colocados em uma proporção menor. Desta forma, quando se faz o uso da roleta é selecionado aleatoriamente um indivíduo para conceber a próxima geração da população, sendo que todos os indivíduos têm chance de reproduzir, porém os mais aptos possuem mais chance de serem selecionados. O processo descrito nesse parágrafo está sendo exemplificado na Figura 2.

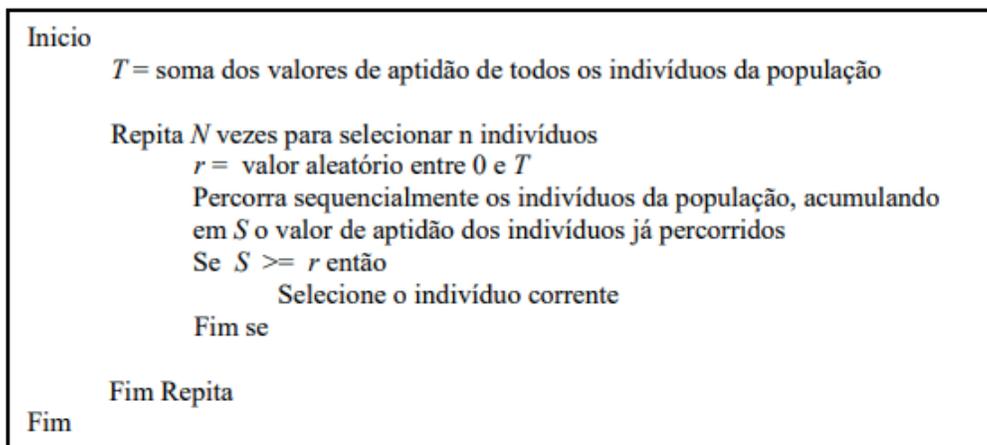


**Indivíduos de uma população e a sua correspondente roleta de seleção.**

**Figura 2** - Exemplo de separação dos indivíduos no método da Roleta. Fonte: CARVALHO (2009)

Com base nestas informações, os autores POZO et al. (2019) apresentam um

pseudocódigo para elaboração do algoritmo da roleta, o qual pode ser observado na Figura 3.



**Figura 3** - Algoritmo Básico do método de seleção por Roleta. Fonte: POZO (2019)

Levando-se em consideração os aspectos apresentados e com a afirmação do autor LUCAS (2002), os algoritmos genéticos são uma ferramenta muito poderosa, principalmente, para problemas de otimização. Sendo um algoritmo robusto, genérico e facilmente adaptável, constitui em uma técnica amplamente estudada e empregada em diversas áreas do conhecimento.

## 1.2 Sistema Imunológico Artificial

O autor Amaral (2006) apresenta que o estudo pelo sistema imunológico humano se desenvolveu, pois os cientistas almejavam poder utilizar esse método para resolução de problemas na área de ciência da computação. O motivo pelo interesse se dá, porque o sistema imunológico evoluiu para fornecer proteção ao organismo contra vários agentes patogênicos. As características de um sistema imunológico que se destacam são:

- Unicidade: cada indivíduo de uma população possui um sistema imunológico diferente;
- Detecção Distribuída: as células do sistema imunológico estão distribuídas por todo o corpo humano e não estão sujeitas a nenhum controle centralizado;
- Aprendizado e memória: o sistema imunológico pode aprender com estruturas de agentes patogênicos para que no futuro uma resposta a este agente ocorra de maneira mais eficiente.
- Detecção Imperfeita: uma detecção completa do agente patogênico não se faz necessária, tornando o sistema flexível;
- Detecção de Anomalias: o sistema imunológico pode detectar e reagir a agen-

tes patogênicos que o corpo humano nunca teve contato anteriormente.

As características apresentadas tornam o sistema imunológico escalável, robusto e flexível. O autor Castro (2006) expõe a complexidade de funcionamento deste sistema: capacidade de reconhecer sinais internos e externos, controlar a ação de componentes de imunidade, influenciar o comportamento de outros sistemas, principalmente sistema nervoso e endócrino, e o aprendizado do combate contra agentes patogênicos. Junto à área de inteligência por partículas, os sistemas imunológicos artificiais constituem os campos mais jovens de pesquisa na área de computação inspirada pela natureza.

Nesse trabalho de pesquisa será utilizado o algoritmo apresentado por Castro e Von Zuben (2002) chamado *Clonal Selection Algorithm* (CLONALG). O algoritmo para resolver problemas de otimização sofre algumas modificações, se comparado ao original, e sua descrição consiste em:

1. Inicialização: Criação de uma população inicial de anticorpos (P);
2. Avaliação de *Fitness*: Determinar o fitness de cada elemento da população inicial;
3. Seleção Clonal e expansão: Selecionar os  $n_1$  elementos com os maiores fitness da população inicial e cloná-los proporcionalmente de acordo com o fitness;
4. Maturação por Afinidade: realizar a mutação em cada clone com uma taxa inversamente proporcional ao fitness. Adicionar os indivíduos que sofrerão mutação na população (P);
5. Metadinâmica: substituir um número  $n_2$  de elementos com os menores fitness e inserir novos elementos que são gerados aleatoriamente;
6. Ciclo: repetir os passos de 2 a 5 até atingir um critério de parada.

Diante do descrito é possível definir o algoritmo CLONALG, que pode ser visualizado na Figura 4:

```
procedimento [P] = CLONALG (max_it, n1, n2)
  inicialização P
  t ← 1
  enquanto t <= max_it faça,
    f ← avaliação (P)
    P1 ← seleção (P, n1, f)
    C ← clonagem (P1, f)
    C1 ← mutação (C, f)
    f1 ← avaliação (C1)
    P1 ← seleção (C1, n1, f1)
    P ← substituição (P, n2)
    t ← t + 1
  fim enquanto
fim procedimento
```

**Figura 4** - Algoritmo de seleção clonal, CLONALG, para otimização. Adaptado de Castro (2006).

### 1.3 Lógica Fuzzy

A teoria dos conjuntos *fuzzy* foi exposta pelo matemático Lotfi Zadeh, com o objetivo de apresentar uma ferramenta matemática para lidar com informações de carácter impreciso ou vago (SANDRI e CORREA, 1999). Na teoria clássica de conjuntos, fica evidente que um elemento pertence ao conjunto que ele está inserido, de tal forma que pode ser expresso por uma função característica observável na Equação 1.

$$f(x) = \begin{cases} 1 & \text{se e somente se } x \in A \\ 0 & \text{se e somente se } x \notin A \end{cases}$$

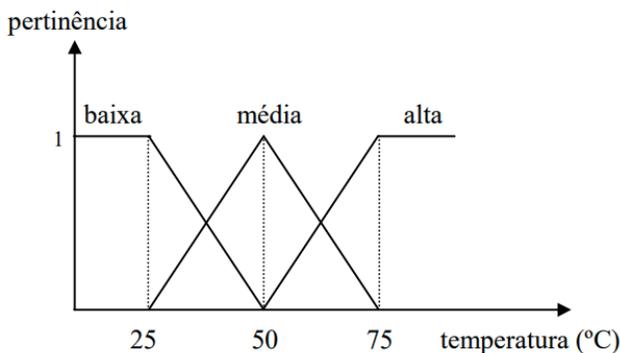
**Equação 1** - Função característica de pertinência de um elemento dentro de um conjunto segundo a teoria clássica. Adaptado de: (TANSCHKEIT, 2004)

Zadeh propôs que a função pudesse assumir infinitos valores no intervalo  $[0,1]$ , desta forma, um elemento dentro de um conjunto passa a possuir um grau de pertencimento no qual é dado pela função pertinência que pode ser visualizado na Equação 2.

$$\mu(x): x \rightarrow [0,1]$$

**Equação 2** - Função pertinência de um elemento dentro de um conjunto *fuzzy* pode resultar num valor dentro do intervalo de zero a um. Adaptado de: TANSCHKEIT (2004)

Uma técnica utilizada para se trabalhar com a lógica *fuzzy* é a das variedades linguísticas. Esta técnica explora a abrangência das diversidades da variável que se deseja atingir com o resultado em termos gramaticais, ou seja, são variáveis não-numéricas que fornecem a representação linguística do problema em uma linguagem natural e/ou artificial (ZIMMERMANN, 2001). Um exemplo prático das variáveis linguísticas seria um sistema *fuzzy* que manipula temperatura; variáveis linguísticas desse processo poderiam ter valores como “baixa”, “média” e “alta”. Estes valores são representados por funções de pertinência, conforme apresentado na Figura 5.



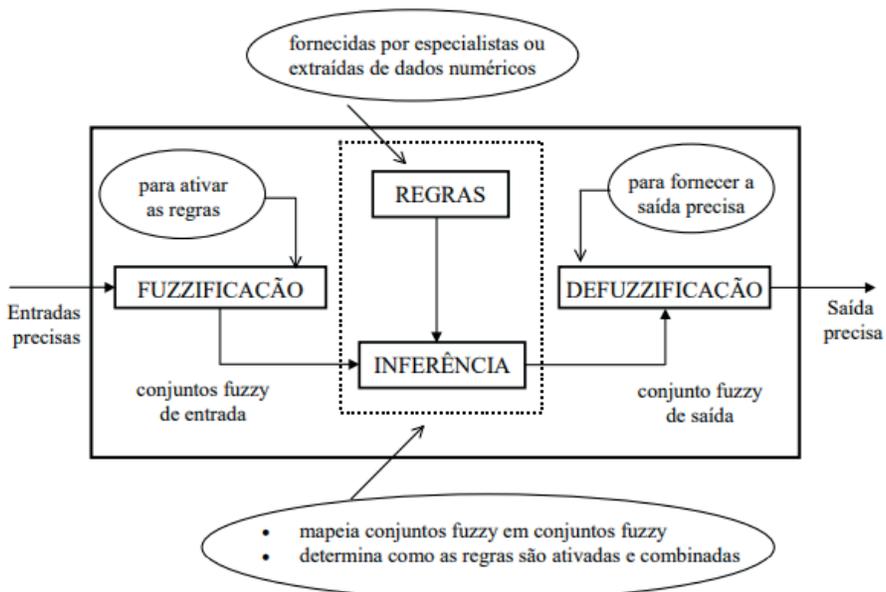
**Figura 5** - Funções de pertinência para a variável temperatura. Fonte: TANSCHKEIT (2004)

Os valores de uma variável linguística são construídos a partir de termos primários (Ex: “alto”, “baixo” e “pequeno”), de conectivos lógicos (Ex: “não”, “e” e “ou”) de modificadores (Ex: “muito”, “pouco”, “extremamente”) e de delimitadores (Ex: parênteses). Como o autor ZIMMERMANN (2001) descreve, uma variedade linguística é definida por uma quintupla  $(N, T(N), X, G, M)$  onde:

- N: nome da variável (Ex: Temperatura);
- T(N): conjunto de termos de N, ou seja, o conjunto de nomes dos valores linguísticos de N. (Ex: {baixa, média, alta});
- X: universo do discurso (Ex:  $-20^{\circ}\text{C}$  a  $130^{\circ}\text{C}$ );
- G: regra sintática para gerar os valores de N como uma composição de termos de T(N), conectivos lógicos, modificadores e delimitadores (Ex: temperatura não baixa e não muito alta.
- Regra semântica, para associar cada valor gerado por G um conjunto *fuzzy* em X.

O autor Tanscheit (2004) explana que as funções de pertinência são definidas a partir de experiências e da perspectiva do usuário, portanto dois sistemas *fuzzy* para uma mesma funcionalidade, provavelmente, conterão funções de pertinência diferentes, uma vez que o contexto é muito relevante para a definição das funções.

Um sistema *fuzzy* possui três etapas denominadas fuzzificação, inferência *fuzzy* e defuzzificação e essas etapas estão demonstradas na Figura 6.



**Figura 6** - Funções de pertinência para a variável temperatura. Fonte: TANSCHIEIT (2004)

Como os autores Souza e Mesquita (2010) descrevem, os sistemas *fuzzy* possuem uma entrada não-*fuzzy*, ou seja, valores exatos/precisos; isso se deve pois, na prática, esses valores, na maioria das vezes, são resultantes de medições ou observações. Desta forma, a etapa de fuzzificação consiste em modelar as entradas dos sistemas por conjuntos *fuzzy*. A etapa de inferência consiste em processar cada proposição *fuzzy* de forma matemática e gerar uma saída respectiva para cada entrada, para que, na etapa de defuzzificação haja um processamento que permita representar o conjunto *fuzzy* novamente em um valor que será interpretado e que representará a saída do sistema.

## 2 | METODOLOGIA

A metodologia utilizada para o desenvolvimento desse projeto de pesquisa consiste na identificação de um problema, sua relevância e, a partir de pesquisas bibliográficas, desenvolver uma solução satisfatória. Toda pesquisa acerca dos conceitos de algoritmos genéticos, sistemas imunológicos artificiais, lógica *fuzzy* e problemas multicritérios e multiobjetivos foram obtidos através de livros, artigos científicos, jornais, revistas, teses e monografias para um estudo consolidado e estruturado em diversos autores da área. A partir das observações e conclusões, foi desenvolvida uma solução para o problema de alocação de benefícios para o quadro de funcionários de uma empresa. Diante do apresentado, foi proposto o desenvolvimento de dois algoritmos para encontrar uma solução ótima do problema. Os instrumentos necessários para a montagem do ambiente de desenvolvimento da pesquisa, visando a produção do algoritmo e a análise dos resultados obtidos, e as características do hardware envolvido estão descritos na Tabela 1.

Descrição de hardware e ferramentas de software utilizados	
Memória RAM	8 GB
Processador	Intel Core i7-4510U
Tipo de Sistema	64 bits
Disco Rígido	27 GB
Sistema Operacional	Ubuntu 18.04.2 LTS
Ambiente de Desenvolvimento	Eclipse for C/C++ developers versão 2019-03
Linguagem de programação	C++, utilizando o paradigma de Orientação à Objetos

**Tabela 1** - Descrição de Hardware e Software. Fonte: Autor Próprio.

Após a etapa da montagem de ambiente, inicia-se o processo de concepção da solução em meio computacional. Primeiramente, foram identificados todos os benefícios que a empresa tem disponível para oferecer aos seus funcionários. Ao realizar essa listagem, levou-se em conta o custo do benefício por funcionário e todas as restrições associadas a estes, como: a quantidade de benefícios que um funcionário pode ter, o

custo máximo que um funcionário pode assumir, se o benefício existente pode ser dado a qualquer funcionário, independente do cargo ou salário, entre outras restrições que podem existir dentro do âmbito empresarial.

Após esta etapa, definiu-se quais as informações pessoais que eram necessárias de cada funcionário e que estavam disponíveis na base de dados da empresa. Tais informações foram utilizadas com a finalidade de descrever a necessidade de cada funcionário para cada benefício; esse processo deu-se utilizando os conceitos de lógica *fuzzy* e foram definidos os estágios de gradação através de termos linguísticos para cada benefício; exemplificando esta etapa do processo tem-se: funcionários com mais de cinquenta anos de idade possuem muita necessidade de um plano de saúde; funcionários com menos de vinte cinco anos possuem pouca necessidade. Vale ressaltar que foram criadas uma tabela para cada cruzamento entre informação pessoal do funcionário e benefício, com a finalidade de identificar como cada característica pessoal influencia na necessidade de se obter um benefício.

Posteriormente, diante de todas as informações geradas até esta fase, gerou-se uma matriz de demanda dos funcionários, ou seja, essa matriz tinha a função de exibir o resultado atingido na quantificação da necessidade de um funcionário por cada benefício. Esse trabalho de pesquisa também propõe a etapa de seleção, ou seja, o momento de definição de qual benefício cada funcionário irá receber, que será realizado utilizando algoritmos genéticos ou um sistema imunológico artificial.

Para se utilizar algoritmos genéticos foi necessário definir os parâmetros do algoritmo, como taxa de mutação, taxa de mortalidade, tamanho da população inicial além do critério de parada que consiste na taxa de similaridade da população. Uma vez definidos esses parâmetros, bastou iniciar o processo de cruzamento dos indivíduos da população até que o critério de parada fosse atendido. Uma vez o critério de parada atingido, esse processo foi encerrado e o resultado obtido pelo algoritmo apresentado. Para a utilização do sistema imunológico artificial também foi necessário definir parâmetros do algoritmo, como tamanho da população inicial, quantidade de clones que são gerados por cada anticorpo, além da taxa de afinidade que indica o momento de encerrar o algoritmo.

Após a definição de todos os parâmetros, a última etapa consistiu em executar ambos os algoritmos e definir métricas de avaliação da performance de cada um. O processo de análise de desempenho dos algoritmos deu-se pelas métricas de tempo que o algoritmo demandou para apresentar um resultado e a qualidade do mesmo, isto é, qual o grau de satisfação média dos funcionários de toda a empresa e o custo mensal da mesma com os investimentos em benefícios. Caso o resultado não fosse satisfatório, seria necessário voltar ao passo de definição dos parâmetros dos algoritmos e executar novamente todos os passos até a apresentação de um resultado aceitável.

Diante do descrito, foi elaborado um algoritmo genético e um sistema imunológico artificial combinados com a técnica de representação de variáveis através de lógica *fuzzy*

com o objetivo de resolver o problema multicritério e multiobjetivo apresentado no seguinte cenário: uma empresa com oitocentos funcionários a qual tem a disposição seis possíveis benefícios - plano de saúde, plano odontológico, auxílio transporte, bolsa de estudo, acesso ao clube da empresa e vale cultura. Cada benefício possui um custo à empresa, além de uma quantidade disponível. As restrições associadas à obtenção dos benefícios estão ligadas à quantidade: um funcionário deve receber no mínimo um benefício e não deve receber mais que três benefícios. Os dados pessoais de cada funcionário que foram analisados são: idade, gênero, distância casa-trabalho, salário, cargo e escolaridade. A partir dos dados pessoais de cada funcionário e de relações pré-estabelecidas entre benefício e característica, foi possível obter uma função satisfação; essa função representa a satisfação de um funcionário diante de uma combinação de benefícios dados a ele; foi determinado que um funcionário se encontra totalmente satisfeito se este recebe os três benefícios mais indicados para ele. Diante disso, foram desenvolvidos os algoritmos de computação natural com o objetivo de selecionar quais são os benefícios mais indicados para cada funcionário receber.

Tanto para o algoritmo genético como para o sistema imunológico artificial, foram criadas uma população inicial; o algoritmo genético contém uma população de tamanho de oitocentos indivíduos, enquanto no sistema imunológico artificial possui apenas quarenta. Em ambos os casos, cada indivíduo desta população contempla uma combinação de benefícios para os oitocentos funcionários da empresa e são gerados de forma aleatória, contudo foi necessário realizar um tratamento após a criação desses indivíduos, com o objetivo de trazê-los para a factibilidade, ou seja, ao criá-los de forma totalmente aleatória, os indivíduos gerados não respeitavam as restrições dadas pela empresa. Em seguida é necessário ordenar a população através da função de aptidão, também conhecida como fitness; tanto o genótipo como os anticorpos, são definidos como uma matriz de booleanos cujo a quantidade de linhas representa a quantidade de funcionários e as colunas os benefícios disponíveis; as interseções entre linhas e colunas indicam se tal funcionário receberá tal benefício. Uma linha desta matriz corresponde a uma combinação de benefícios para o funcionário e a partir dela, pode-se calcular a satisfação do mesmo. Desta forma, a métrica da função fitness consiste na satisfação média da solução, que é calculada através do somatório da satisfação de cada funcionário. A função *fitness* também leva em consideração dois fatores: custo e penalidades; o custo de cada benefício é informado ao início da execução e assim é possível obter o custo final da solução e, por fim, as penalidades estão associadas às restrições que não são cumpridas por uma proposta de solução, ou seja, caso um funcionário receba mais que três benefícios ou nenhum, isto indica uma penalidade; caso a solução utilize mais benefícios que o disponível a penalidade também é acrescida e caso o custo da solução seja maior que o custo máximo parametrizado; ao final estas penalidades adicionam uma constante no custo final.

Para o algoritmo genético, após a geração da população inicial e a classificação

da mesma, iniciou-se o processo de cruzamento. Os pais foram selecionados através do método da roleta e a geração de um novo indivíduo foi dada da seguinte maneira: o novo indivíduo é constituído dos genes dos pais, portanto neste caso, os genes foram selecionados de forma aleatório entre os dois pais e cada gene representa uma linha da matriz que consiste na combinação de benefícios para um funcionário. Ao gerar um filho, esse é inserido na população de forma classificada. A mutação dos filhos, também está presente; os filhos que foram selecionados para mutação sofrem algumas alterações em seus genes: alguns genes são selecionados e os elementos que os constituem são alterados de posição de forma que um benefício é deixado de ser recebido em detrimento de outro que passou a ser obtido. O critério de parada constitui em três possíveis casos: o primeiro se o máximo de iterações foi atingido, sendo este um parâmetro do algoritmo; o segundo, se a solução permaneceu a mesma sem alteração do melhor indivíduo durante dez por cento do máximo de iterações e o terceiro caso se a população atingiu um grau de similaridade, que também consiste em um parâmetro configurável.

Para o sistema imunológico artificial, após a criação da população inicial e a classificação da mesma, iniciou-se o processo de mutação; cada indivíduo da população gerou um valor parametrizado de novos indivíduos os quais foram classificados e ao final restaram na população a quantidade inicial de indivíduos antes do processo, em resumo, o tamanho da população é constante. O processo de mutação de cada indivíduo (anticorpo), deve ser inversamente proporcional a satisfação, ou seja, quanto mais próxima a solução atual está da solução ótima, menor o número de representações de empregados é selecionado para sofrer mutações. A mutação consistiu num processo de três possibilidades: adição de um novo benefício, exclusão de um benefício já recebido ou troca de benefícios. Ao desejar adicionar um benefício para um empregado é verificado se este recebe menos que três benefícios, devido a restrição, bem como se a restrição de custo total da solução já não foi atingido; após, é verificado se existem benefícios disponíveis e se os benefícios disponíveis não são os mesmos que o emprego já recebe e caso todas as condições sejam satisfeitas o emprego passa a receber um novo benefício; caso seja detectado que ele não possa receber um benefício é tentado fazer com que ele troque um deles. O processo de troca consiste em sortear outro empregado e realizar, se possível, a troca de um benefício entre eles; caso não seja possível, é realizada a exclusão de um benefício. A exclusão de um benefício se dá caso o empregado já recebe mais que dois benefícios e, de maneira aleatória, é excluído um deles. Uma condição inserida que teve grande impacto na performance e nas soluções encontradas foi que um empregado somente sofreu mutação caso sua satisfação média estivesse menor do que a satisfação média da solução como um todo. Os mesmos critérios de paradas aplicados ao algoritmo genético foram aplicados ao sistema imunológico artificial.

A interface do algoritmo ocorre toda através de arquivos JSON; para iniciar o algoritmo, este deve receber três arquivos que contém respectivamente os dados pessoais

dos empregados, os parâmetros dos algoritmos e um terceiro arquivo contendo a média salarial e o desvio padrão. Como resultado são gerados outros dois arquivos, também no formato JSON, que constituem na solução encontrada pelo algoritmo; o primeiro arquivo contém os resultados da solução de forma mais geral com informações de custo total, tempo de execução, satisfação média, quantidade utilizada de cada benefício e a quantidade de empregados em cada faixa de porcentagem de satisfação; o outro arquivo contém informações detalhadas de cada funcionário como os benefícios que ele deve receber e o custo associado, qual a ordem de preferência para receber os benefícios e a satisfação que ele atingiu com os benefícios entregues. Uma vez encontrado um resultado e todos os arquivos JSON criados, para melhorar a visibilidade e análise dos dados, foi criada uma interface com um sistema web. O sistema web foi desenvolvido com o objetivo de trazer facilidade através de uma interface amigável ao usuário.

O sistema web desenvolvido, foi dividido em cinco telas: *dashboard*, gerenciador de parâmetros, gerenciador de funcionários, solução e solução completa; cada uma delas possui uma determinada função. A Figura 7 apresenta a tela de *dashboard* do sistema, onde é possível iniciar a execução do software e visualizar os logs em tempo real.



**Figura 7** - Tela de dashboard sistema web. Fonte: Autor Próprio.

A tela do gerenciador, que pode ser visualizada na Figura 8, é responsável por permitir que o usuário altere os parâmetros de execução do *software*, os parâmetros gerais, os parâmetros do algoritmo genético e os parâmetros do sistema imunológico artificial.

Parâmetros Gerais			
Descrição	Valor default	Parâmetro	Valor default
Quantidade de Benefícios	6	Quantidade disponível do Plano de Saúde	600
Custo do Plano de Saúde	200	Quantidade disponível do Plano Odontológico	650
Custo do Plano Odontológico	180	Quantidade disponível do Auxílio Transporte	570

**Figura 8** - Tela do gerenciador do sistema web. Fonte: Autor Próprio.

A terceira tela consiste no gerenciador de funcionários e possui como premissa ser uma tela de consulta, porém devido a necessidade de facilitar a usabilidade do usuário foram atribuídas novas funcionalidades: importar os funcionários através de um CSV, adicionar ou deletar um funcionário, gerar o JSON de funcionários a ser consumido pelo software através do sistema e visualizar o JSON gerado; esta tela pode ser observada na Figura 9.

**Funcionários**

[Home](#) > [Funcionários](#)

Importar Funcionários:

Nenhum arquivo selecionado

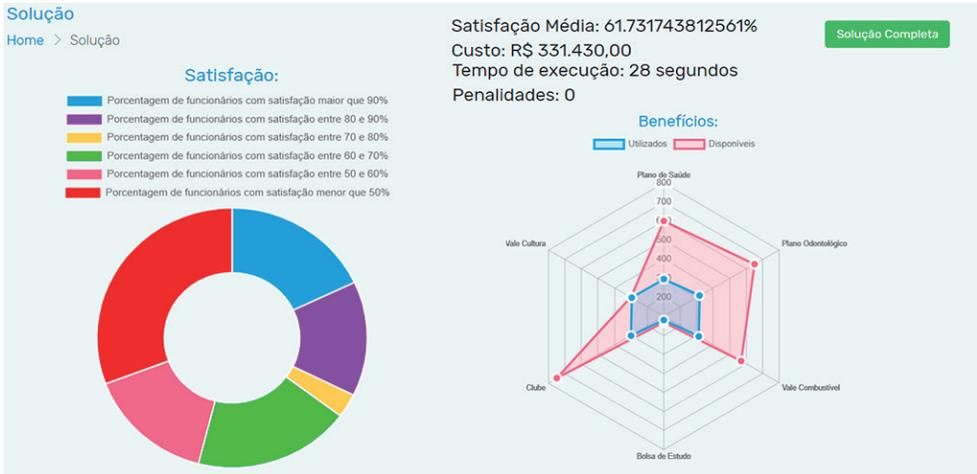
---

Funcionários

#	Idade	Gênero	Distância do Trabalho	Salário	Cargo	Escolaridade	Ação
9	64	M	51.59	6765.75	Senior	Tecnico	<input type="button" value="Deletar"/>
10	40	M	39.23	3774.13	Junior	Ensino fundamental	<input type="button" value="Deletar"/>

**Figura 9** - Tela dos funcionários sistema web. Fonte: Autor Próprio.

A tela de solução, apresentada na Figura 10, exibe uma solução geral após a execução do algoritmo, e as informações apresentadas são: gráfico de satisfação, gráfico de distribuição de benefícios, satisfação média atingida, custo da solução, tempo de execução do algoritmo e quantidade de penalidades.



**Figura 10** - Tela da solução do sistema web. Fonte: Autor Próprio.

Por fim, a tela da solução completa, apresentada conforme a Figura 11, apresenta a solução individual de cada funcionário, onde é possível visualizar o ID do funcionário, o custo individual desse funcionário para empresa, as ordens de preferência dos benefícios, os benefícios recebidos e a satisfação individual do funcionário.

Solução NaturalUSF

- Plano de Saúde
- Vale Combustível
- Bolsa de Estudo
- Plano Odontológico
- Clube
- Vale Cultura

#	Custo	Preferência	Recebido	Satisfação
10000	R\$ 363,00			56.42 %
10001	R\$ 473,00			97.04 %
10002	R\$ 513,00			98.05 %
10003	R\$ 343,00			57.71 %

**Figura 11** - Tela da solução completa do sistema web

Para o desenvolvimento do sistema web foram utilizadas as tecnologias mais atuais e difundidas no mercado, no qual para o *front-end* foi utilizado a biblioteca CSS conhecida como *Bootstrap* e para o desenvolvimento do *back-end* foi utilizado a linguagem de programação PHP. Para a base de dados dos funcionários foi utilizado a linguagem MySQL. O *Bootstrap* é um kit de ferramentas de open source para desenvolvimento com HTML, CSS e JS. Seu desenvolvimento é responsivo, além de possuir componentes pré-construídos em CSS e plugins criados no jQuery. Originalmente criado por um *designer* e desenvolvedor do Twitter, o *Bootstrap* se tornou uma das estruturas de *front-end* e projetos de código aberto mais populares do mundo. Atualmente é mantida por um grupo de

desenvolvedores no GitHub. (BOOTSTRAP, 2019). O PHP é uma linguagem de script *open source* de uso geral, muito utilizada, e é especialmente adequada para o desenvolvimento web e que pode ser embutida dentro do HTML. A diferença do PHP para o JavaScript no lado do cliente, é que o código é executado no servidor, gerando o HTML que é então enviado para o navegador. O navegador recebe os resultados da execução desse *script*, mas não sabe qual era o código fonte. (PHP, 2019). O MySQL é um Sistema de Gestão de Banco de Dados (SGDB) relacional conhecido como o mais popular do mundo dentre as soluções de bancos de dados de código aberto (*open source*). Suas características são: alta velocidade, capacidade de executar várias threads simultaneamente, multiusuário e por fim uma estrutura robusta, que garante a integridade das informações armazenadas (MySQL AB, 2011).

### 3 | RESULTADOS E DISCUSSÃO

Para avaliar o desempenho dos algoritmos desenvolvidos, o primeiro passo foi criar uma massa de dados referente aos oitocentos funcionários de uma empresa, a quantidade e o valor referente a cada benefício além de definir os parâmetros iniciais para cada um dos algoritmos. A Tabela 2 exibe os valores utilizados relacionados aos benefícios - quantidade e o custo.

Parâmetro	Valor	Parâmetro	Valor
Quantidade de planos de saúde e custo associado	[600, R\$ 200]	Quantidade de planos odontológicos e custo associado	[650, R\$ 180]
Quantidade de bolsas de estudo e custo associado	[130, R\$ 500]	Quantidade de associados ao clube e custo associado	[750, R\$ 143]
Quantidade de auxílio transporte e custo associado	[570, R\$ 150]	Quantidade de vales cultura e custo associado	[300, R\$ 220]

**Tabela 2** - Parâmetros utilizados com relação aos benefícios Fonte: Autor Próprio.

De modo a comparar a efetividade dos códigos elaborados foram executados os algoritmos desenvolvidos (algoritmo genético e sistema imunológico artificial) para a mesma massa de dados dez vezes. Os resultados obtidos com a execução do algoritmo genético podem ser visualizados na Tabela 3.

N	Satisfação	Custo [R\$]	Iterações	Tempo de Execução [s]	Critério de Parada
1	95,7220 %	455.413,00	3500	1005	Máximo de Iterações sem alteração do melhor indivíduo.
2	95,7317 %	454.519,00	5000	1738	Máximo de Iterações atingido.
3	95,5530%	449.600,00	5000	1710	Máximo de Iterações atingido.
4	95,0706%	448.394,00	5000	1661	Máximo de Iterações atingido.
5	96,2446%	450.433,00	3400	848	Máximo de Iterações sem alteração do melhor indivíduo.
6	95,0940%	453.589,00	3200	914	Máximo de Iterações sem alteração do melhor indivíduo.
7	96,0658%	453.105,00	4300	1124	Máximo de Iterações sem alteração do melhor indivíduo.
8	95,4506%	455.437,00	3300	882	Máximo de Iterações sem alteração do melhor indivíduo.
9	96,0013%	447.611,00	3500	929	Máximo de Iterações sem alteração do melhor indivíduo.
10	95,2955%	449.853,00	3100	806	Máximo de Iterações sem alteração do melhor indivíduo.

**Tabela 3** - Resultados obtidos com a execução do algoritmo genético Fonte: Autor Próprio.

A primeira coluna da tabela constitui o identificador da execução e as seguintes colunas representam, respectivamente, satisfação média da solução, custo pela empresa da solução, quantidade de gerações/iterações, tempo de execução em segundos e o critério de parada utilizado. Diante dos dados apresentados, pode-se perceber que o algoritmo está convergindo para uma solução ótima, pois após dez execuções, o nível de satisfação atingiu na média 95,62% com um desvio padrão de 0,00383; o custo atingiu uma média de R\$ 451.795,40 o que representa um custo médio de R\$ 564,74 para cada funcionário e o tempo médio de execução foi de 1161 segundos.

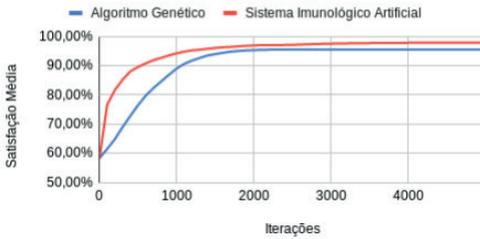
Da mesma maneira, foram executados os mesmos testes com o sistema imunológico artificial, realizando dez execuções do algoritmo, inclusive com a mesma massa de dados utilizado pelo algoritmo genético e os resultados obtidos são demonstrados na Tabela 4 que mantém a mesma estrutura já apresentada.

N	Satisfação	Custo [R\$]	Iterações	Tempo de Execução [s]	Critério de Parada
1	97,8801 %	460.911,00	5000	1204	Máximo de Iterações atingido.
2	96,0088 %	460.137,00	1900	1240	Máximo de Iterações sem alteração do melhor indivíduo.
3	94,2647%	461.332,00	1600	1315	Máximo de Iterações sem alteração do melhor indivíduo.
4	93,5214%	460.703,00	1400	1379	Máximo de Iterações sem alteração do melhor indivíduo.
5	98,0018%	461.446,00	3400	1853	Máximo de Iterações sem alteração do melhor indivíduo.
6	95,5094%	460.630,00	1900	1273	Máximo de Iterações sem alteração do melhor indivíduo.
7	98,0573%	461.861,00	5000	1111	Máximo de Iterações atingido.
8	97,7565%	459.986,00	5000	1058	Máximo de Iterações atingido.
9	97,9107%	461.336,00	5000	1099	Máximo de Iterações atingido.
10	97,5585%	460.010,00	3000	1471	Máximo de Iterações sem alteração do melhor indivíduo.

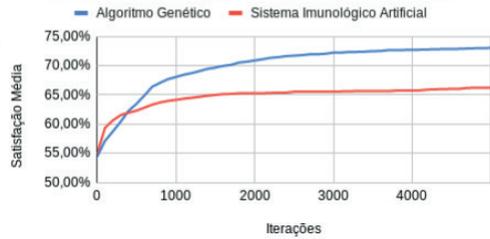
**Tabela 4** - Resultados obtidos com a execução do sistema imunológico artificial. Fonte: Autor Próprio.

Diante dos dados apresentados, pode-se perceber que o algoritmo está convergindo para uma solução ótima, pois após dez execuções, o nível de satisfação atingiu na média 97,65% com um desvio padrão de 0,01858; o custo atingiu uma média de R\$ 460.807,00 o que representa um custo médio de R\$ 576,00 para cada funcionário e o tempo médio de execução foi de 1256 segundos. Em seguida, foi realizado outro teste, que consistiu em adicionar uma restrição monetária para verificar o comportamento dos algoritmos. A restrição adicionada foi de R\$ 300.000,00 e neste cenário o algoritmo genético foi muito superior ao sistema imunológico artificial; os resultados obtidos foram: ambos algoritmos executaram o máximo de iterações permitidas que consistiu em cinco mil iterações; o algoritmo genético apresentou uma solução após 526 segundos, com uma satisfação média de 73,03% e um custo de R\$ 299.832,00 enquanto que o sistema imunológico artificial apresentou uma solução após 10891 segundos com uma satisfação média de 66,22% e um custo de R\$ 299.896,00. Como forma de comparar a efetividade dos algoritmos, construiu-se o gráfico apresentado na Figura 12, o qual apresenta a relação entre satisfação média atingida e quantidade de iterações de cada algoritmo; o gráfico A representa os casos descritos nas tabelas 3 e 4, enquanto que o gráfico B apresenta o cenário descrito com restrição monetária.

**A - Comparativo do desempenho dos algoritmos através do número de iterações**



**B - Comparativo do desempenho dos algoritmos através do número de iterações com restrição monetária**



**Figura 12** - Gráficos que relacionam a satisfação média com a quantidade de iterações para cada algoritmo desenvolvido. Fonte: Autor próprio.

Diante das informações obtidas, é possível observar que quando as restrições dos algoritmos são folgadas, ambos apresentaram um alto índice de satisfação e baixo desvio padrão, demonstrando a efetividade dos códigos desenvolvidos em atingir uma solução muito próxima do ideal, desta forma, mostraram-se eficientes ferramentas na seleção de problemas multicritérios e multiobjetivos, utilizando um tempo consideravelmente razoável, para um problema que seria considerado complexo para resolução devido ao seu tamanho e quantidade de variáveis e restrições. O sistema imunológico artificial apresentou resultados levemente melhores se comparado ao algoritmo genético, contudo o genético apresentou maior precisão no resultado, uma vez que seu desvio padrão é menor. Também é possível observar que após duas mil iterações, em ambos os casos, praticamente a solução já está obtida, portanto o tempo despendido pelo algoritmo pode ser reduzido mais da metade, pois após este intervalo, é realizado um ajuste fino. Contudo, quando foi adicionado uma restrição monetária, relacionada ao custo máximo, o algoritmo genético apresentou um desempenho muito superior se comparado ao sistema imunológico artificial principalmente com relação ao tempo; desta forma, ao considerarmos ambos os cenários testados, o algoritmo genético apresentou o melhor desempenho.

## 4 | CONCLUSÃO

Em virtude dos fatos descritos nesse artigo é possível concluir que os algoritmos genéticos e os sistemas imunológicos artificiais se mostraram ferramentas efetivas e adequadas na solução do problema multicritério e multiobjetivo proposto que consistiu na alocação de benefícios para um quadro de funcionários. As soluções apresentadas pelos algoritmos desenvolvidos atingiram uma solução bem próxima ao ideal em todas as suas execuções e, além disso, se mostraram precisos, como indica o desvio padrão obtido. O uso da lógica *fuzzy* contribui positivamente para o resultado atingido, pois o seu uso facilitou a representação e manipulação dos dados pessoais dos funcionários, resultando numa maior assertividade no momento da criação da função que calcula a satisfação de cada

colaborador. Devido à natureza do problema, o resultado esperado era que o algoritmo genético tivesse um desempenho melhor se comparado ao sistema imunológico artificial, contudo ambos os algoritmos se mostraram eficientes quando as restrições eram folgadas; alterando-se os parâmetros dos benefícios, ambos continuaram eficientes, contudo ao alterar o parâmetro referente ao custo máximo da solução, o algoritmo genético mostrou um desempenho muito superior ao sistema imunológico artificial, mostrando assim, que de fato, o algoritmo genético representa uma importante ferramenta para solução de problemas cuja formulação matemática seja definida através do uso de programação inteira e combinatória, sobretudo em problemas cujos critérios e objetivos sejam pluri.

## REFERÊNCIAS

BOOTSTRAP. **ABOUT: Learn more about the team maintaining Bootstrap, how and why the project started, and how to get involved..** [S. l.], 2019. Disponível em: <https://getbootstrap.com/docs/4.3/about/overview/>. Acesso em: 5 nov. 2019.

CARVALHO, André Ponce de Leon F. de. **Algoritmos Genéticos**. 2009. 1 f. Tese (Doutorado) - Curso de Engenharia de Computação, Núcleo de Computação Eletrônica, University Of São Paulo, São Carlos, 2009. Disponível em: <http://conteudo.icmc.usp.br/pessoas/andre/research/genetic/>. Acesso em: 08 mar. 2019.

CASTRO, Leandro N. de et al. **Computação Natural: Uma Breve Visão Geral**. Disponível em: [https://www.researchgate.net/publication/261438133\\_Computacao\\_Natural\\_Uma\\_Breve\\_Visao\\_Geral](https://www.researchgate.net/publication/261438133_Computacao_Natural_Uma_Breve_Visao_Geral). Acesso em: 21 mar. 2019.

CASTRO, Leandro Nunes de. **Fundamentals of Natural Computing: Basic Concepts, Algorithms, and Applications**. Boca Raton: Chapman & Francis Group, 2006. 662 p.

CASTRO, Leandro Nunes de; VON ZUBEN Fernando J. 2002. **Learning and Optimization Using the Clonal Selection Principle**, IEEE Transactions on Evolutionary Computation, 6(3), pp.239-251. Disponível em: [https://pdfs.semanticscholar.org/29ce/df9f9edd5179aa2e78654a7b81c9da45f0d0.pdf?\\_ga=2.213258960.537033840.1574027227-578011725.1574027227](https://pdfs.semanticscholar.org/29ce/df9f9edd5179aa2e78654a7b81c9da45f0d0.pdf?_ga=2.213258960.537033840.1574027227-578011725.1574027227). Acesso em: 21 mar. 2019.

COELLO, Carlos A. Coello. **Evolutionary multi-objective optimization: a historical view of the field**. 2006. Disponível em: <https://ieeexplore.ieee.org/document/1597059>. Acesso em: 20 abr. 2019.

GOEDERT, Matheus L.; PAULA FILHO, Pedro L.; BLANCO, Daniel R.. **Computação natural: conceitos e aplicações da computação inspirada na natureza**. Espacios, v. 38, 2017. Disponível em: <http://www.revistaespacios.com/a17v38n34/a17v38n34p31.pdf>. Acesso em: 21 mar. 2019.

GOMES, Carlos Francisco Simões; COSTA, Helder Gomes. **Aplicação de métodos multicritério ao problema de escolha de modelos de pagamento eletrônico por cartão de crédito**. Production, São Paulo, v. 25, n. 1, p.54-68, jan. 2015. Disponível em: <http://www.scielo.br/pdf/prod/v25n1/0103-6513-prod-0103-6513-2014-056412.pdf>. Acesso em: 21 mar. 2019.

GOMIDE, Fernando A. C.; GUDWIN, Ricardo R.; TANSCHKEIT, Ricardo. **Conceitos Fundamentais da Teoria de Conjuntos Fuzzy, Lógica Fuzzy e Aplicações**. Disponível em: <ftp://vm1-dca.fee.unicamp.br/pub/docs/gudwin/publications/ifs95.pdf>. Acesso em: 21 mar. 2019.

LUCAS, Diogo C.. **Algoritmos Genéticos: uma Introdução**. 2002. Disponível em: <<http://www.inf.ufsc.br/~luis.alvares/INE5633/ApostilaAlgoritmosGeneticos.pdf>>. Acesso em: 09 mar. 2019.

MOTA, Tiago da Conceição. **Introdução a Algoritmos Genéticos**. 2007. Disponível em: <<https://docplayer.com.br/29396305-Introducao-a-algoritmos-geneticos.html>>. Acesso em: 09 mar. 2019  
MySQL AB. **MySQL Reference Manual**. [S. l.], 2001. Disponível em: [http://www.ic.unicamp.br/~celio/livrobd/mysql/mysql\\_manual.pdf](http://www.ic.unicamp.br/~celio/livrobd/mysql/mysql_manual.pdf). Acesso em: 5 nov. 2019.

PETROLI NETO, Sílvio. **Computação Evolutiva: desvendando os algoritmos genéticos**. 2011. Disponível em: <<http://www.portal.anchieta.br/revistas-e-livros/ubiquidade/pdf/artigo4.pdf>>. Acesso em: 09 mar. 2019.

PHP. **O QUE é o PHP?**. [S. l.], 2019. Disponível em: [https://www.php.net/manual/pt\\_BR/intro-what-is.php](https://www.php.net/manual/pt_BR/intro-what-is.php). Acesso em: 5 nov. 2019.

POZO, Aurora et al. **Computação Evolutiva**. Disponível em: <<http://www.inf.ufpr.br/aurora/tutoriais/Ceapostila.pdf>>. Acesso em: 09 mar. 2019

SANDRI, S.; CORREA, C. **Lógica Nebulosa**, 1999. Disponível em: <[https://www.gta.ufrj.br/ensino/cpe717-2011/curso\\_ERN99\\_fuzzy.pdf](https://www.gta.ufrj.br/ensino/cpe717-2011/curso_ERN99_fuzzy.pdf)>. Acesso em: 07 Março 2019.

SILVA, Ricardo Fernandes da; PIGNATA, Maria Izabel Barnez. **CHARLES DARWIN E A TEORIA DA EVOLUÇÃO**. Disponível em: <<https://www.cepae.ufg.br/up/80/o/TCEM2014-Biologia-RicardoFernandesSilva.pdf>>. Acesso em: 09 mar. 2019

SOUZA, Osmar do Nascimento; MESQUITA, Marcos Eduardo R. Valle. **Introdução à Teoria dos Conjuntos Fuzzy**. 2010. Disponível em: <<http://www.uel.br/pessoal/valle/PDFfiles/osmar10.pdf>>. Acesso em: 15 out. 2019.

TANSCHKEIT, Ricardo. **Sistemas Fuzzy**, 2004. Disponível em: <<http://www.inf.ufsc.br/~mauro.roisenberg/ine5377/leituras/ICA-Sistemas%20Fuzzy.pdf>>. Acesso em: 15 out. 2019.

ZAMBIASI, Prof. Dr. Saulo Popov. **Algoritmos Genéticos**. 2011. 1 f. Tese (Doutorado) - Curso de Engenharia de Computação, Universidade Federal de Santa Catarina (UFSC)., Santa Catarina, 2011. Disponível em: <<https://www.gsigma.ufsc.br/~popov/aulas/ia/modulo10/index.html>>. Acesso em: 09 mar. 2019.

ZIMMERMANN, H.-J. **Fuzzy Set Theory—and Its Applications**. 4ª. ed. [S.l.]: Springer Science+Business Media New York, 2001. Disponível em: <<https://books.google.com.br/books?hl=pt-BR&lr=&id=HVHtCAAQBAJ&oi=fnd&pg=PR13&dq=fuzzy+logic+theory+sets&ots=sfYhV9Vu44&sig=NtZLjRbUbaEZXWuyZbXZfOU4CTk#v=onepage&q&f=false>>. Acesso em: 08 Março 2019.