



CENTRO UNIVERSITÁRIO LUTERANO DE PALMAS

Recredenciado pela Portaria Ministerial nº 1.162, de 13/10/16, D.O.U. nº 198, de 14/10/2016
AELBRA EDUCAÇÃO SUPERIOR - GRADUAÇÃO E PÓS-GRADUAÇÃO S.A.

Larisse Rodrigues Alves

UTILIZAÇÃO DE ALGORITMOS GENÉTICOS NA ORGANIZAÇÃO DE GRUPOS DE TRABALHO EM SALA DE AULA

Palmas - TO

2021

Larisse Rodrigues Alves

UTILIZAÇÃO DE ALGORITMOS GENÉTICOS NA ORGANIZAÇÃO DE GRUPOS DE
TRABALHO EM SALA DE AULA

Dissertação elaborada e apresentada como requisito parcial para aprovação na disciplina de Trabalho de Conclusão de Curso II (TCC II) do curso de bacharel em Sistemas de Informação pelo Centro Universitário Luterano de Palmas (CEULP/ULBRA).

Orientador: Prof. M.e Fabiano Fagundes.

Palmas - TO

2021

Larisse Rodrigues Alves
UTILIZAÇÃO DE ALGORITMOS GENÉTICOS NA ORGANIZAÇÃO DE GRUPOS DE
TRABALHO EM SALA DE AULA

Dissertação elaborada e apresentada como requisito parcial para aprovação na disciplina de Trabalho de Conclusão de Curso II (TCC II) do curso de bacharel em Sistemas de Informação pelo Centro Universitário Luterano de Palmas (CEULP/ULBRA).

Orientador: Prof. M.e Fabiano Fagundes.

Aprovado em: ____/____/____

BANCA EXAMINADORA

Prof. M.e Fabiano Fagundes

Orientador

Centro Universitário Luterano de Palmas – CEULP

Prof^a. Esp. Fernanda Pereira Gomes

Centro Universitário Luterano de Palmas – CEULP

Prof. Esp. Fábio Castro Araújo

Centro Universitário Luterano de Palmas – CEULP

Palmas - TO

2021

RESUMO

ALVES, Larisse Rodrigues. **Utilização de Algoritmos Genéticos na organização de grupos de trabalho em sala de aula**. 2021. 41 f. Trabalho de Conclusão de Curso (Graduação) – Curso de Sistemas de Informação, Centro Universitário Luterano de Palmas, Palmas/TO, 2021¹.

A sala de aula é um ambiente composto por diversos perfis de estudantes com diferentes ritmos de aprendizagem. A combinação destas particularidades em grupos de trabalho permite o compartilhamento de conhecimento individual e o desenvolvimento de habilidades de comunicação e convivência. Combinar adequadamente os diversos perfis de alunos, alinhado com os objetivos da atividade proposta é um desafio para professor/instrutor responsável. Sendo assim, este trabalho resultou no desenvolvimento de uma ferramenta capaz de automatizar o processo de agrupamento de estudantes, baseado em Algoritmos Genéticos. Apresentando resultados favoráveis para formação de grupos homogêneos e mostrando que o modelo proposto necessita de ajustes para organização de grupos heterogêneos.

PALAVRAS-CHAVE: Algoritmos Genéticos, Trabalhos em Grupo, Formação de Grupos.

¹ Elemento incluído com a finalidade de posterior publicação do resumo na internet. Sua formatação segue a norma ABNT NBR 6023, por isto o alinhamento e o espaçamento diferem do padrão do texto.

LISTA DE FIGURAS

| | |
|--|----|
| Figura 1 - Estrutura da Computação Natural | 10 |
| Figura 2 - Estrutura da Inteligência Computacional | 11 |
| Figura 3 - Estrutura da Computação Evolutiva | 12 |
| Figura 4 - Funcionamento Básico dos AGs | 13 |
| Figura 5 - Codificação Binária | 16 |
| Figura 6 - Codificação por Permutação | 16 |
| Figura 7 - Codificação em Árvore | 16 |
| Figura 8 - Codificação de Valores | 17 |
| Figura 9 - Seleção por Roleta | 17 |
| Figura 10 - Crossover - Codificação em Árvore | 19 |
| Figura 11 - Etapas do Desenvolvimento | 23 |
| Figura 12 - Interface para cadastro das informações dos alunos | 25 |
| Figura 13 - Interface para cadastro das informações dos grupos | 26 |
| Figura 14- Representação binária dos alunos | 27 |
| Figura 15- Codificação Binária - Resultados | 29 |
| Figura 16 - Exemplo do método roleta | 31 |
| Figura 17 - Resultado do teste do AG desenvolvido | 34 |
| Figura 18 - Resultado do teste do AG desenvolvido - Alunos Divergentes | 36 |

LISTA DE TABELAS

| | |
|--|----|
| Tabela 1 - Pseudocódigo AGs | 14 |
| Tabela 2 - Terminologias | 14 |
| Tabela 3 - Pseudocódigo – Seleção por Roleta | 17 |
| Tabela 4 - Métodos do AG desenvolvido | 24 |
| Tabela 5 - Exemplo de informações cadastradas na ferramenta | 26 |
| Tabela 6 - Demonstração das listas auxiliares | 27 |
| Tabela 7 - Representação de um Indivíduo (cromossomo) | 27 |
| Tabela 8 - Indivíduo gerado a partir de uma função aleatória | 28 |
| Tabela 9 - População - conjunto de indivíduos | 29 |
| Tabela 10 - Funcionamento da Função Avaliação - Etapa 01 | 30 |
| Tabela 11 - Funcionamento da Função Avaliação - Etapa 02 | 30 |
| Tabela 12 - Pseudocódigo Função Seleção | 31 |
| Tabela 13 - Pseudocódigo Função Cruzamento | 32 |

LISTA DE ABREVIATURAS E SIGLAS

CN – Computação Natural

IC – Inteligência Computacional

CP – Computação Evolutiva

AGs – Algoritmos Genéticos

SUMÁRIO

| | | |
|---------|--|-----------|
| 1 | INTRODUÇÃO | 9 |
| 2. | REFERENCIAL TEÓRICO | 10 |
| 2.1 | COMPUTAÇÃO NATURAL | 10 |
| 2.1.1 | Algoritmos Genéticos | 13 |
| 2.1.1.1 | Codificação de Indivíduos | 15 |
| 2.1.1.2 | Seleção de Indivíduos para Próxima Geração | 17 |
| 2.1.1.3 | Operadores Genéticos | 18 |
| 2.2 | FORMAÇÃO DE GRUPOS | 20 |
| 3. | MATERIAIS E MÉTODOS | 23 |
| 3.1 | MATERIAIS | 23 |
| 3.2 | MÉTODOS | 23 |
| 4. | RESULTADOS E DISCUSSÃO | 25 |
| 4.1 | ENTRADA DE DADOS | 25 |
| 4.2 | MODELAGEM E CODIFICAÇÃO DO PROBLEMA | 26 |
| 5. | CONSIDERAÇÕES FINAIS | 37 |
| 6. | REFERÊNCIAS | 38 |

1 INTRODUÇÃO

O conceito de trabalho em grupo pode ser entendido como a colaboração entre duas ou mais pessoas para concluir uma tarefa ou atribuição, no cenário de sala de aula permite que os alunos interajam e contribuam uns com os outros (STUDY, 2017). A formação de grupos quando realizada de forma adequada impacta positivamente na aprendizagem, permite a troca de conhecimento e colabora para o desenvolvimento de habilidades de comunicação e convivência.

Ao se adotar a prática de atividades colaborativas bem elaboradas, é possível diminuir a carga de trabalho atribuída aos alunos e proporcionar o aumento de produtividade na realização das atividades propostas, além de contribuir para o aprimoramento de aptidões como, a capacidade de gerenciar e resolver problemas de forma eficiente (ZHAMRI et al., 2010). Entretanto, para um agrupamento eficaz e capaz de maximizar os benefícios do trabalho em grupo é necessário que os professores definam critérios (OUNNAS, 2010) e utilizem um método de abordagem (MIRANDA et al., 2017).

O processo de agrupar alunos quando realizado manualmente considerando critérios e abordagem torna-se complexo, por envolver diversas variáveis que caracterizam os alunos, tais como, suas competências, habilidades interpessoais, conhecimentos e outras, adiciona-se ainda a esta lista, quando necessário, as características desejadas ao grupo pelo professor. Exigindo assim, que seja feito o uso de esforço intelectual para relacionar todas as variáveis descritas anteriormente, de forma a gerar grupos consistentes para o desenrolar das atividades.

Dessa forma, visando a criação de grupos adequados para trabalhos realizados em sala de aula de forma automatizada, ou seja, utilizando esforço computacional, o presente trabalho propõe o desenvolvimento de uma ferramenta de agrupamento baseado na abordagem de Algoritmos Genéticos (AGs). Estes Algoritmos simulam o processo de Evolução das Espécies descrito por Darwin (1859) e, através das técnicas de seleção, *crossover* e mutação, permite o desenvolvimento de métodos meta-heurísticos.

As técnicas dos Algoritmos Evolutivos AGs permitem trabalhar com problemas de otimização combinatória dentro de um grupo finito de possíveis soluções em busca das melhores combinações (TSURUTA e NARCISO, 2001). Sendo assim, no contexto deste trabalho os AGs permitiram através da exploração combinatória trabalhar e alinhar as diversas características dos alunos em busca de soluções de agrupamento adequados e, que cumpram as restrições desejadas para o grupo.

2. REFERENCIAL TEÓRICO

Esta seção apresenta a fundamentação teórica utilizada na construção deste trabalho. A subseção 2.1, expõe os conceitos relevantes sobre a área de Computação Natural e suas ramificações até Algoritmos Genéticos; seguido da subseção 2.2 que aborda os conceitos relativos à Formação de Grupos.

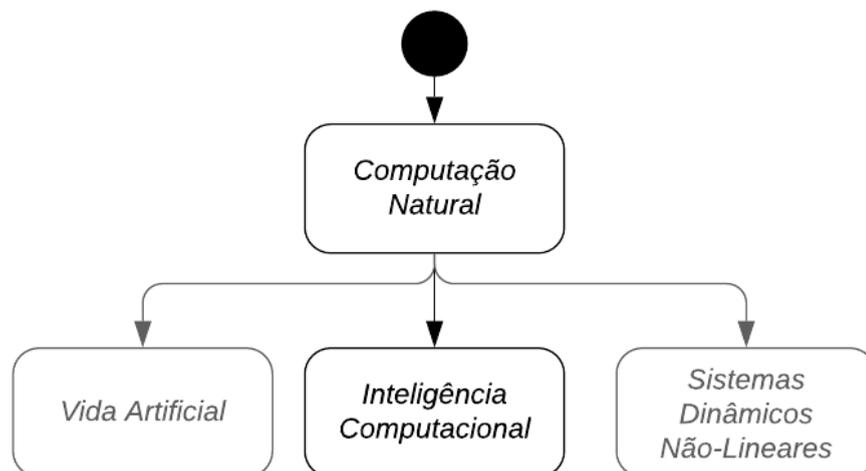
COMPUTAÇÃO NATURAL

A Computação Natural (CN) é a área da computação que busca aplicar e desenvolver novas tecnologias capazes de resolver problemas complexos utilizando como inspiração a natureza. Para Blanco, Paula Filho e Goedert (2017) a CN é um elo entre a Ciência da Computação e as Ciências Naturais. Castro et al. (2004) apresentam alguns objetivos da CN, sendo:

- desenvolver ferramentas matemáticas para a solução de problemas complexos;
- abstrair e projetar sistemas e fenômenos naturais;
- produzir novas formas de vida, apresentada pelos autores como vida artificial;
- utilizar mecanismos naturais como novos paradigmas da computação para suplementar e/ou complementar as tecnologias atuais.

A CN se ramifica em três áreas, conforme apresentado na Figura 1:

Figura 1 - Estrutura da Computação Natural

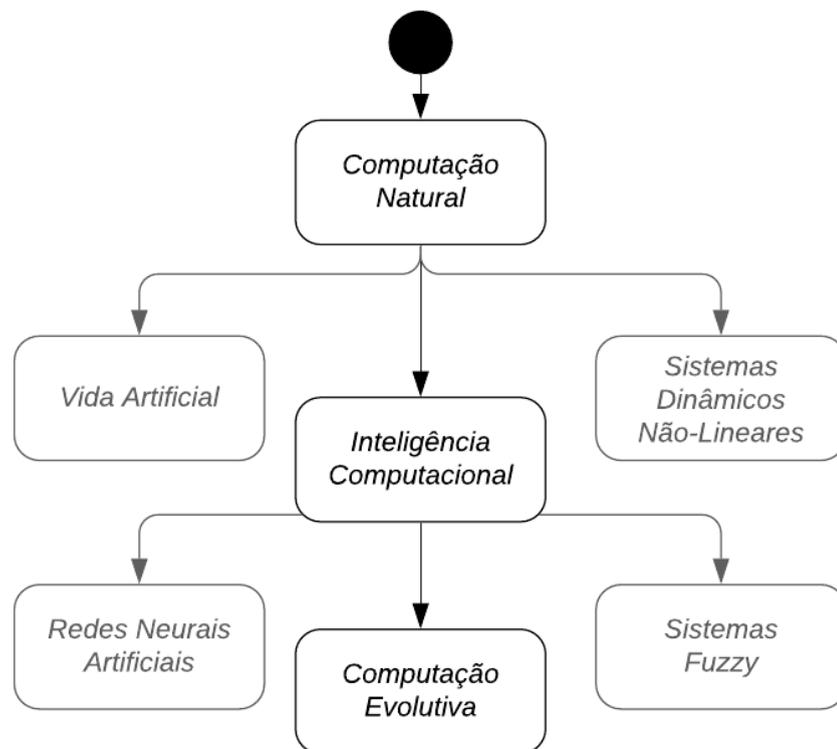


Aprofundando nas ramificações da CN, apresentadas na Figura 1, o próximo conceito a ser abordado é o da Inteligência Computacional (IC), esta subárea da Computação Natural segundo Goldschmidt (2010) busca simular o comportamento humano em atividades específicas, enquanto para Iyoda (2000) a área de IC compreende paradigmas para reproduzir

ou se aproximar de alguma maneira da inteligência/raciocínio de determinados sistemas biológicos.

O IEEE ([20--?]) - *Computational Intelligence Society*, define Inteligência Computacional como “a teoria, projeto, aplicação e desenvolvimento de paradigmas computacionais motivados biologicamente e linguisticamente” que possui três pilares tradicionais: Redes Neurais, Sistemas *Fuzzy* e Computação Evolutiva, que ao longo dos anos foi se ampliando e passou a envolver outras áreas como, Inteligência Ambiental, Vida Artificial etc. As subáreas da IC são apresentadas na Figura 2.

Figura 2 - Estrutura da Inteligência Computacional



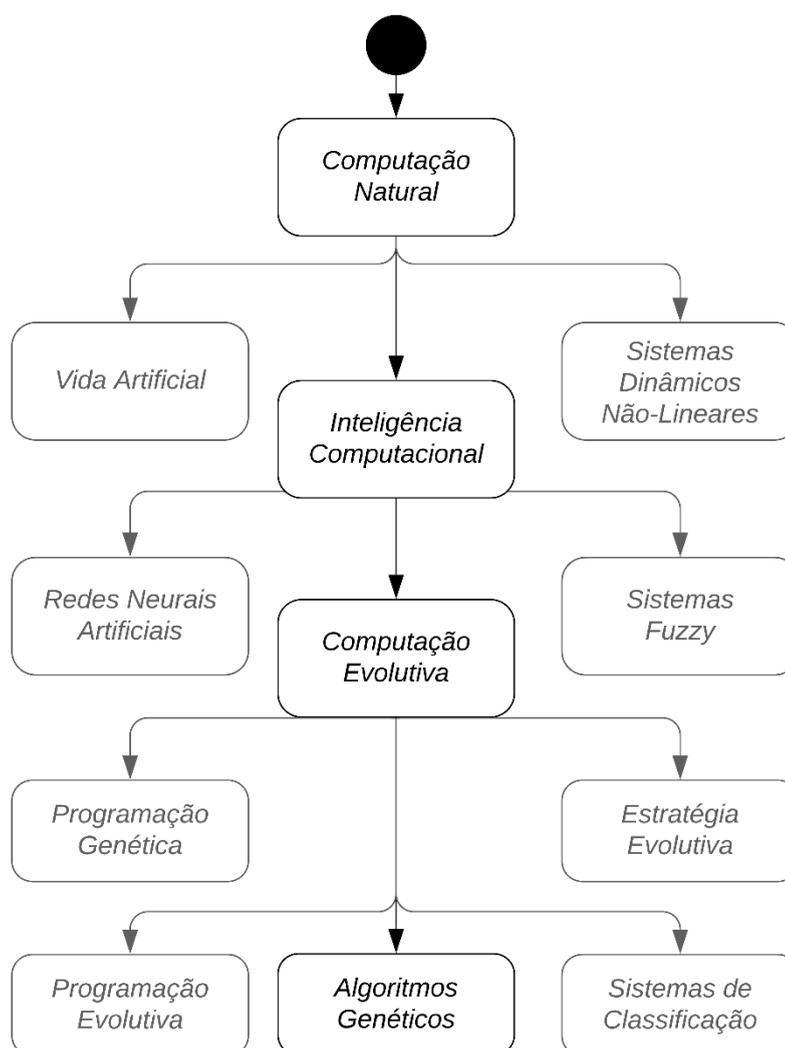
Dentre as subáreas da Inteligência Computacional o próximo contexto a ser compreendido é o de Computação Evolutiva (CE), definido como um novo paradigma que busca solucionar problemas através de técnicas da evolução natural, estes baseados no princípio da teoria evolutiva Darwin (ZUBEN, 2000; POZO et al., 2005).

Os mecanismos naturais repetidos pela CE envolvem as técnicas de reprodução, mutação, recombinação, seleção natural e sobrevivência. Com um conjunto de comandos computacionais (funções) gera-se uma população, neste caso, um conjunto de possíveis soluções que serão avaliados a partir de uma função de aptidão para que se selecione os melhores candidatos e assim originar a próxima geração, através de operadores de recombinação e mutação (EIBEN e SCHOENAUER, 2002).

As subáreas de CE compreendem segundo Pila (2015) os Algoritmos Genéticos (AG), Programação Genética (PG), Estratégias Evolutivas (EE), Programação Evolutiva (PE) e os Sistemas de Classificação (SC) que apesar de serem distintas, possuem modelagem semelhante de acordo com a teoria de Darwin. Contudo, Zuben (2000) destaca que as principais diferenças entre as ramificações da CE são os operadores genéticos utilizados.

A figura a seguir, ilustra as ramificações existentes na CE.

Figura 3 - Estrutura da Computação Evolutiva



Assim, é possível compreender como a computação busca, através de áreas como a Computação Natural, desenvolver tecnologias inspiradas na natureza e com capacidade para se ramificar de acordo com seus objetivos, sendo capazes de simular computacionalmente habilidades de raciocínio e inteligência através da subárea de Inteligência Computacional, que por sua vez, originam novos paradigmas computacionais baseados na teoria da evolução das espécies de Charlie Darwin por meio da Computação Evolutiva. Por fim, seguindo a estrutura

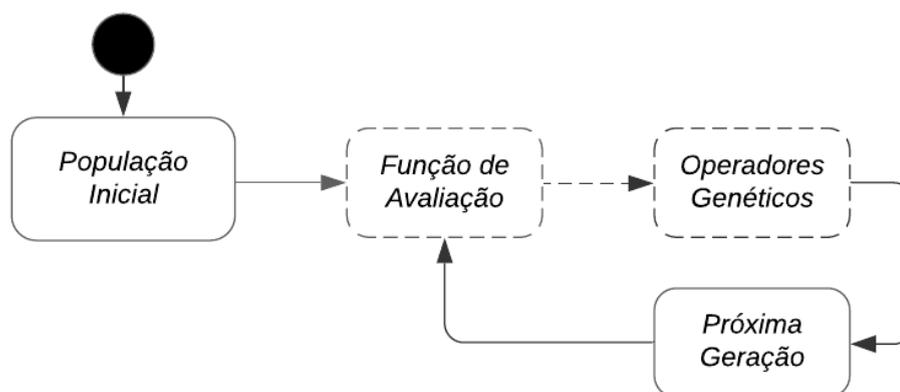
da CN e suas ramificações, é alcançada a área de Algoritmos Genéticos (AGs), conceito fundamental para o desenvolvimento deste trabalho, que terá seus principais conceitos abordados na subseção seguinte.

1.1.1 Algoritmos Genéticos

A teoria da evolução das espécies de Charles Darwin que descreve o organismo mais bem adaptado como sobrevivente da seleção natural inspirou o desenvolvimento dos algoritmos Genéticos (AGs), modelo heurístico eficiente desenvolvido na década de 60 por John Holland, para solucionar problemas difíceis ainda não resolvidos computacionalmente. Este modelo se popularizou por volta de 1975, quando John Holland, apresentou um modelo de AGs estruturado em binário que buscavam abstrair o processo de seleção natural e reproduzir computacionalmente (LIDEN, 2008).

Holland (1992) define AGs como programas de computadores capazes de evoluir de forma semelhante à seleção natural e solucionar problemas complexos através da sua capacidade de explorar inúmeras possibilidades de solução. Linden (2008) destaca ainda a capacidade dos AGs para trabalhar com a mesma população inicial em busca de inúmeras soluções, de maneira direcionada através de mecanismos de seleção e sem necessitar de informações externas. A Figura a seguir ilustra o funcionamento básico dos AGs.

Figura 4 - Funcionamento Básico dos AGs



Fonte: Traduzido de Moreno, Ovalle e Vicari (2012)

Para reproduzir o modelo de seleção natural com AGs, conforme apresentado na Figura 4 é necessário que exista uma população inicial. Realiza-se então, o processo de avaliar e identificar os melhores indivíduos, seguido da aplicação de operadores genéticos responsáveis pela recombinação e mutação desta população (MORENO; OVALLE; VICARI, 2012).

A tabela a seguir apresenta a esquematização da estrutura básica dos AGs em formato de pseudocódigo.

Tabela 1 - Pseudocódigo AGs

1. Inicializar a população

Faça enquanto

2. Avaliar a população
3. Selecionar os indivíduos mais aptos
4. Cruzar os indivíduos
5. Mutar os indivíduos
6. Nova população

Condição de parada atendida? → *Não* → **Retorna ao segundo passo**

↓

7. Apresenta a melhor solução

Fonte: Adaptado de Medeiros (2018)

Assim é possível identificar como a teoria de seleção natural de Darwin se relaciona com os AGs. Em uma população inicial existirá seres mais bem adaptados que, neste caso serão, sobreviventes selecionados de forma natural dependendo do ambiente em que vivem e que se reproduzem, favorecendo assim, o processo de mutação e resultando na evolução das espécies. Os AGs replicam esses comportamentos através de funções para criar uma população, avaliar (*fitness*) e selecionar os indivíduos mais adequados à solução do problema, aplicando em seguida os operadores genéticos responsáveis por realizar o processo de cruzamento (*crossover*) e mutação dos indivíduos.

Para melhor entendimento sobre o tema, é necessário compreender como as terminologias utilizadas na biologia referente à evolução genética são representadas em uma aplicação que utiliza AGs. É possível compreender essa relação observando a Tabela 1 elaborada de acordo com Linden (2008).

Tabela 2 - Terminologias

| Linguagem Natural | Algoritmos Genéticos |
|-------------------|---|
| Cromossomo | Indivíduo, <i>string</i> , cromossomo, árvore |
| Gene | Característica |
| Alelo | Valor |
| <i>Locus</i> | Posição |
| Genótipo | Estrutura |

| | |
|----------|------------------------|
| Fenótipo | Conjunto de parâmetros |
|----------|------------------------|

Fonte: Linden (2008)

Moreno, Ovalle e Vicare (2012) apresentam um indivíduo, neste caso um cromossomo como uma solução viável ou não para um problema onde cada gene (característica) é uma porção desta possível solução. Assim, se houver uma população muito pequena a ser trabalhada pelo Algoritmo Genético é possível que não exista uma diversidade necessária para se encontrar uma boa solução para o problema, por outro lado, uma grande população pode resultar na ineficiência do AG, que poderá passar muito tempo explorando as inúmeras possibilidades de resultados (URSEM, 2002).

Zuben (2000) considera que para utilizar o AG em um dado problema deve-se seguir as seguintes etapas:

- definir o processo de representação genética denominado codificação;
- método para criar/inicializar uma população: geralmente inicializada aleatoriamente que, de forma geral, resulta em uma boa distribuição e exploração das soluções (indivíduos) no espaço de busca;
- uma forma de avaliar os indivíduos: avaliados através de uma função de aptidão, conhecida também como função *fitness*. Esta determina a qualidade de um indivíduo e o atribui um valor que represente o quão bom ou ruim esta solução é para um problema em questão;
- um método de seleção, *crossover* e mutação.

A seguir, as etapas de codificação, seleção, *crossover* e mutação serão detalhadas de acordo com Obitko (1998).

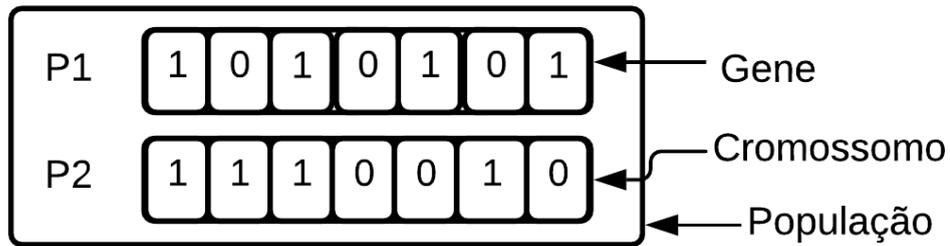
1.1.1.1 Codificação de Indivíduos

A codificação é a parte fundamental para as etapas seguintes. A escolha adequada de como codificar o problema interfere diretamente no desempenho do AG a ser desenvolvido.

Os principais métodos de codificação envolvem:

- codificação binária: o modelo mais simples e mais utilizado. Uma população é o agrupamento de vários cromossomos, estes representados por um conjunto de bits (0 ou 1). Muitas vezes a codificação binária pode não ser a melhor escolha, sendo necessário fazer correções antes de se aplicar os operadores genéticos. A representação da codificação binária é apresentada na Figura 5 a seguir.

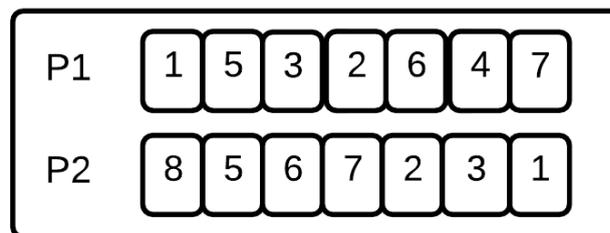
Figura 5 - Codificação Binária



Fonte: Adaptado de Obitko (1998)

- codificação por permutação: utiliza-se em problemas de ordenação. Cada cromossomo é uma série de números que representam uma posição em uma sequência. Para aplicação dos métodos de cruzamento e mutação é preciso se atentar, para que não ocorra alterações nos cromossomos. O modelo desta codificação é apresentado na Figura 6 a seguir.

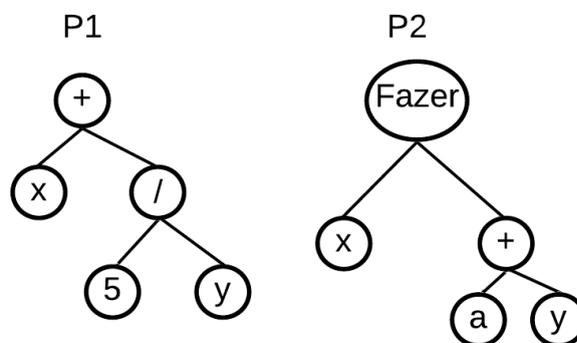
Figura 6 - Codificação por Permutação



Fonte: Adaptado de Obitko (1998)

- codificação em árvore: cada cromossomo neste método representa uma árvore com alguns objetos, que podem ser comandos ou funções de uma linguagem de programação, que devem seguir uma sequência lógica de execução, conforme apresentado na Figura 7. A aplicação dos operadores genéticos é realizada de maneira mais fácil.

Figura 7 - Codificação em Árvore



Fonte: Adaptado de Obitko (1998)

- codificação de valores: nesta técnica cada cromossomo é representado por um conjunto de valores que podem ser complexos para codificação binária, os indivíduos podem ser representados por números reais, *strings* etc., conforme apresentado na Figura 8.

Figura 8 - Codificação de Valores



Fonte: Adaptado de Obitko (1998)

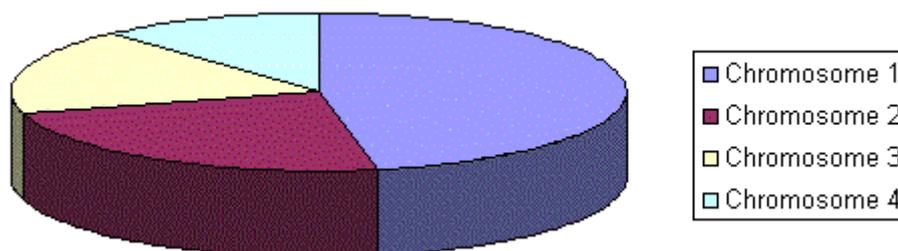
Quando utilizado este tipo de codificação é necessário que se desenvolva um método para aplicação dos operadores genéticos.

1.1.1.2 Seleção de Indivíduos para Próxima Geração

Para realizar o processo de seleção é necessário inicialmente atribuir um valor de aptidão a cada indivíduo através de uma função denominada *fitness*. Esta nota permitirá ao método de seleção avaliar e identificar os mais aptos, estes serão escolhidos e recombinados a fim de formar a próxima geração. Para isto, utiliza-se das seguintes técnicas:

- seleção por roleta: representa a ideia de que em uma circunferência, similar a uma roleta onde serão dispostos todos os cromossomos da população e, cada um terá um pedaço correspondente ao seu valor de aptidão, uma bolinha é lançada para selecionar um indivíduo, sendo assim, o cromossomo com maior pedaço é o mais provável de ser escolhido. Uma técnica que apresenta problemas quando a grande variação entre os valores de adequação dos indivíduos. A Figura 9, representa a seleção por roleta.

Figura 9 - Seleção por Roleta



Fonte: Obitko (1998)

O pseudocódigo deste método, é apresentado na Tabela 3 a seguir.

Tabela 3 - Pseudocódigo – Seleção por Roleta

-
1. [**soma**] some os valores de adequação de todos os cromossomos da população – soma **S**
 2. [**seleção**] gere um número aleatório no intervalo de $(0, S)$ – seleção **R**
 3. [**repetição**] percorra toda a população e some os valores de adequação (soma **S**) quando **S** for maior que **R** pare e retorne o cromossomo selecionado.

Obs.: o passo 1 é realizado uma única vez por população

Fonte: Adaptado de Obitko (1998)

- seleção por classificação: inicialmente a população é classificada e aos cromossomos são atribuídos valores de adequação, determinados de acordo com sua classificação. Permitindo assim que todos os indivíduos possam ser selecionados em algum momento.
- seleção por estado estacionário: este método busca formar uma nova geração com cromossomos de desempenho excelente, ou seja, alta adequação. Sendo estes selecionados para formar a próxima geração e em seguida os cromossomos com desempenhos menores são removidos da população e substituídos.
- elitismo: os melhores cromossomos são selecionados para iniciar a nova população e o resto dos indivíduos são selecionados através do processo de cruzamento e seleção. Este método aumenta o desempenho dos AGs por prevenir a perda da melhor solução encontrada.

O processo de seleção deve privilegiar os indivíduos com avaliação alta para gerar descendentes, mas sem desprezar os menos aptos, já que até mesmo estes podem ter características genéticas que sejam favoráveis para se chegar a melhor solução.

1.1.1.3 Operadores Genéticos

A aplicação dos operadores genéticos (cruzamento e mutação) está diretamente ligado aos métodos de codificação escolhidos para o problema. Sendo assim, para a codificação binária o processo de cruzamento (*crossover*) pode ser:

- *crossover* de ponto único: um ponto de cruzamento é definido, a série binária do começo do cromossomo até o ponto definido é copiada do primeiro pai e o resto copiado do segundo pai, resultando assim no próximo indivíduo. A seguir, têm-se a representação deste processo realçado em negrito.

$$11001011 + 11011111 = 11001111$$

- *crossover* de dois pontos: são definidos dois pontos de cruzamento. O primeiro pai é responsável por passar seus genes do início de seu cromossomo até o primeiro ponto de cruzamento definido e do segundo ponto até o final. O segundo pai compartilha com seu descendente os seus genes localizados a partir do primeiro ponto de partida até o segundo, conforme exemplificado a seguir:

$$11001011 + 11011111 = 11011111$$

- *crossover* uniforme: os bits são copiados aleatoriamente dos dois pais e replicados no descendente, conforme apresentado em seguida.

$$11001011 + 11011101 = 11011111$$

- cruzamento aritmético: é realizada uma operação lógica, neste caso, *and* para formar os descendentes, o processo descrito é exemplificado em seguida.

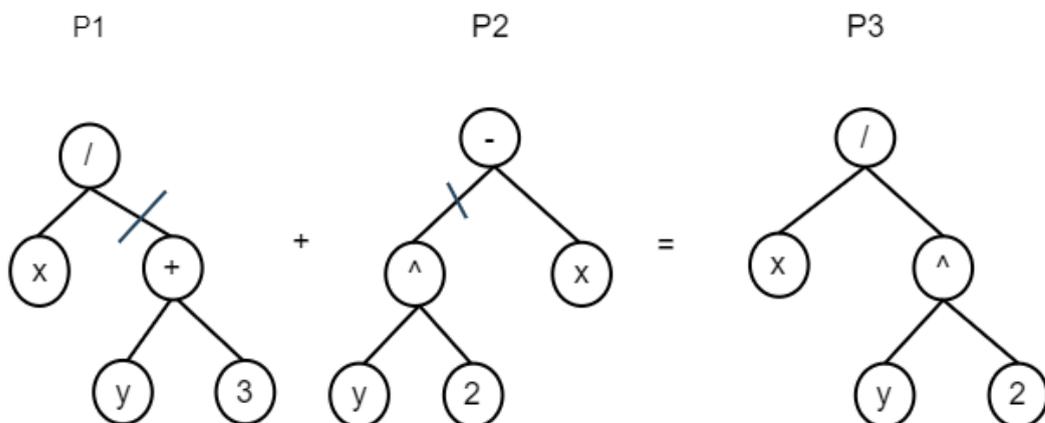
$$11001011 + 11011111 = 11001011$$

O processo de mutação para a codificação binária é relativamente simples e envolve somente a inversão dos bits dos cromossomos. Para o método codificação de valores, todos os cruzamentos utilizados na codificação binária podem ser aplicados, mas para realizar o processo de mutação deve-se basear em um gene específico, onde pode-se adicionar ou retirar um pequeno valor. Conforme exemplificado a seguir, nos valores destacados em negrito.

$$(1.29 \ 5.68 \ \mathbf{2.86} \ \mathbf{4.11} \ 5.55) \Rightarrow (1.29 \ 5.68 \ \mathbf{2.73} \ \mathbf{4.22} \ 5.55)$$

A codificação em árvore, apresentada na Figura 10 é semelhante ao cruzamento de ponto único.

Figura 10 - Crossover - Codificação em Árvore



Fonte: Adaptado de Obitko (1998)

Nesta codificação é necessário definir um ponto de cruzamento nos dois pais, os nós iniciais até o ponto de cruzamento são copiados do primeiro pai e a partir deste ponto, os nós do segundo pai são trocados e reproduzidos no descendente.

Ao se trabalhar a formação de grupos, seja de trabalho ou de estudo, pode-se desejar obter a melhor configuração para atingir um determinado objetivo. O uso dos Algoritmos Genéticos pode auxiliar neste processo, ao permitir que sejam organizadas diferentes configurações de grupos, com diferentes características, gerando ao final o grupo (ou grupos) mais aptos, através da exploração combinatória das características e/ou habilidade dos alunos. A seguir, são apresentadas as características da formação de grupos que poderão ser utilizadas ao organizar a estruturação dos grupos a serem analisados via AG.

1.2 FORMAÇÃO DE GRUPOS

A definição de grupo pode ser entendida como duas ou mais pessoas envolvidas no mesmo propósito e capazes de interagir entre si (GENÇER, 2019). A aprendizagem cooperativa permite a melhoria contínua do processo de conhecimento dos integrantes do grupo de forma individual e coletiva, além de promover o apoio mútuo, incentivo e o encorajamento uns aos outros para desenvolvimento das atividades (JOHNSON; JOHNSON; SMITH, 1998).

De acordo com Johnson e Johnson (2009) o trabalho em grupo promove o surgimento dos seguintes elementos básicos:

- interdependência positiva: representa a ligação adquirida entre os membros dos grupos, que percebem a necessidade de trabalhar em conjunto para alcançar o objetivo proposto;
- responsabilidade individual: com o desempenho pessoal afetando diretamente os resultados do grupo, os integrantes passam a sentir necessidade de serem responsáveis e de contribuírem de forma individual e efetiva para garantir o bem-estar do grupo;
- interação promotora: os membros dos grupos promovem a produtividade uns dos outros para o desenvolvimento das atividades;
- habilidades interpessoais e de pequenos grupos: os integrantes dos grupos trabalham as habilidades cooperativas relacionadas a liderança, tomada de decisão, construção de confiança, comunicação e gerenciamento de conflitos;
- processamento de grupo: corresponde a necessidade dos integrantes de um grupo de discutirem o quão bem eles estão alcançando seus objetivos e mantendo

relações de trabalho eficazes entre os membros.

Contudo, para formação de grupos educacionais eficientes e que gerem bons resultados, Ounnas (2010) considera que é preciso definir critérios e escolher cuidadosamente a abordagem a ser utilizada para divisão dos grupos. O processo de agrupamento tem como problema inicial a definição de como os alunos serão distribuídos aos grupos (SRBA e BIELIKOVA, 2015).

Ounnas, Davis e Millard (2007) definem três principais abordagens que geralmente são utilizadas para realizar às divisões de grupos, sendo estas:

- abordagem auto selecionada: agrupamento negociado ou realizado pelos próprios estudantes;
- abordagem selecionada: os grupos são formados de acordo com algum critério previamente definido pelo professor;
- abordagem aleatória: o processo de agrupamento é realizado pelo professor, mas sem critérios específicos.

As abordagens, aleatórias e auto selecionada, segundo Srba e Bielikova (2015) apresentam sérios problemas para formação de grupos. Segundo os autores, estas abordagens podem gerar grupos altamente desequilibrados e que tenham resultados ineficientes, ou em grupos homogêneos que tendem a impedir a disseminação de conhecimentos e ideias entre os alunos e, até mesmo, resultar no isolamento de alguns estudantes.

A combinação de alunos realizada manualmente e considerando critérios, neste caso a abordagem selecionada, tem como desvantagem o seu nível de complexidade, que consiste na tentativa de combinar os diversos perfis de alunos com base em características, habilidades ou informações previamente adquiridas (SRBA e BIELIKOVA, 2015).

O processo de formação de grupo pode ser classificado de acordo com as seguintes formas: automatizada ou manual. A divisão de grupos quando realizada pelo professor sem auxílio de sistemas entende-se que foi utilizado a forma manual, caso o agrupamento seja realizado com a contribuição de um sistema denomina-se automatizada.

O agrupamento pode ainda considerar alguns critérios como, homogeneidade, grupos formados por alunos com características semelhantes, heterogeneidade e/ou misto (MIRANDA et al., 2017). Para definição de grupos com base nestes critérios podem-se considerar as habilidades de competências técnica, habilidades interpessoais (MELLO; ARIMA; NEVES, 2014) ou os aspectos cognitivos, interesse, personalidade, papéis a serem desempenhados no grupo, sexo, idade etc. (CAETANO, 2013, p.58, apud ROMNEY, 1996).

Os critérios de agrupamento dos estudantes devem estar alinhados aos resultados que se esperam alcançar com a atividade. Para abordagens que proponham o desenvolvimento elevado das capacidades e habilidades dos alunos de níveis básico e alto, deve-se considerar o agrupamento heterogêneo que promove o convívio com os diferentes perfis existentes em sala. Contudo, os alunos medianos são mais bem incentivados quando estão agrupados de forma heterogênea.

O estudo realizado por Marzano, Pickering e Pollock (2004) aponta que a formação de grupos homogêneos tem efeito variado de acordo com os tipos de alunos agrupados. Os resultados apresentados pelos autores mostram que, alunos com baixo domínio de um conteúdo, se agrupados com colegas semelhantes não se desenvolvem durante o decorrer das atividades, enquanto os grupos de alunos com alto potencial se beneficiaram positivamente, mas de forma pouco significativa. Por outro lado, o estudo indica que os aprendizes de média capacidade apresentam os melhores resultados com este tipo de formação, e os grupos heterogêneos favorecem os alunos com dificuldade a desenvolverem suas habilidades.

1. MATERIAIS E MÉTODOS

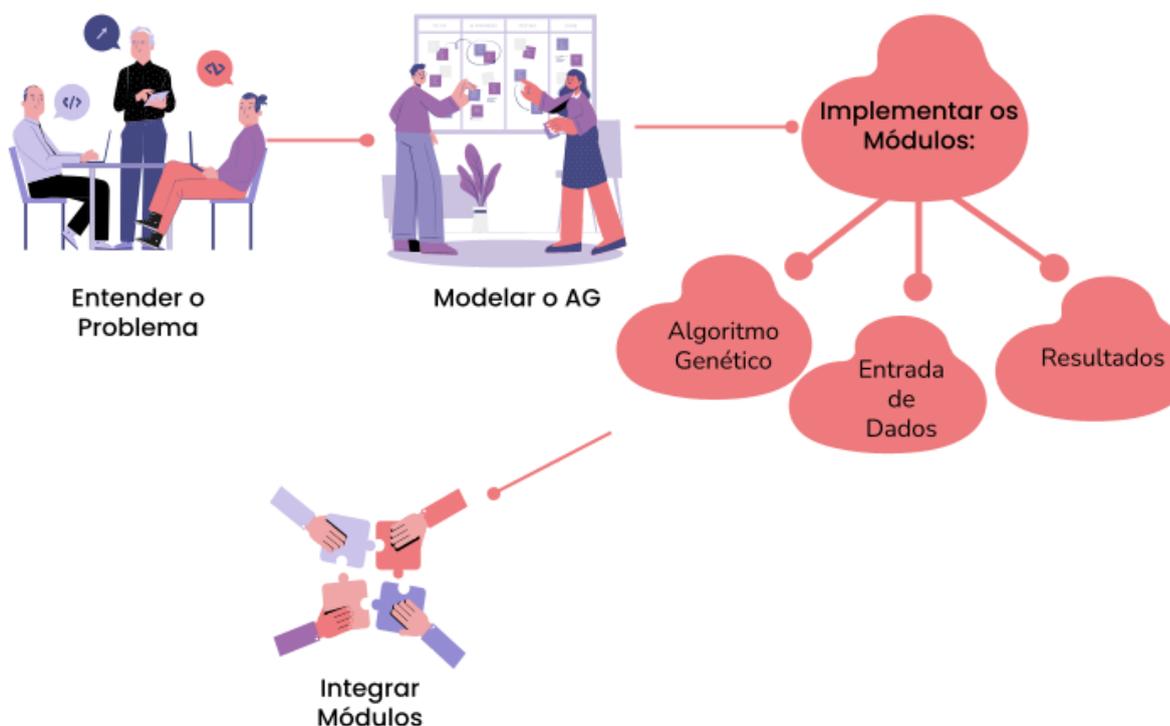
2.1 MATERIAIS

As tecnologias utilizadas no desenvolvimento deste trabalho envolveram o uso da linguagem de programação de alto nível Python que possui diversas bibliotecas que auxiliam no desenvolvimento de aplicações de *Machine Learning* e *IA* e que foi utilizada em conjunto com o *Framework Django*, que facilita o desenvolvimento de aplicações *web*. Sendo assim, a linguagem de programação *Python* foi utilizada para o desenvolvimento do Algoritmo Genético e no processamento dos dados, enquanto o *Framework web Django* permitiu a integração do Algoritmo Genético desenvolvido a uma *interface* de interação com o usuário, responsável por receber os dados de entrada e apresentar os resultados.

2.2 MÉTODOS

O desenvolvimento deste trabalho foi realizado em etapas, conforme ilustrado na figura abaixo.

Figura 11 - Etapas do Desenvolvimento



As etapas apresentadas na figura anterior serão detalhadas a seguir:

Inicialmente junto ao especialista do domínio foram definidas a abordagem e o método de agrupamento que foram utilizados, conforme indicado na subseção 2.2, formação de

grupos. Ainda nesta etapa, foram determinados os dados de entrada referentes aos alunos e aos grupos aceitos pelo AG. Assim, verificou-se a necessidade de, para cada aluno, informar nome e até três características e, aos grupos, atribuir um nome identificador, a característica desejada e o limite de integrantes.

Após entender o problema, definiu-se o método de codificação, seleção e os operadores genéticos a serem utilizados. A tabela a seguir, apresenta a relação entre os métodos utilizados no decorrer deste trabalho e as etapas do AG.

Tabela 4 - Métodos do AG desenvolvido

| Etapas do AG | Modelo Desenvolvido |
|--|---|
| Codificação | Binária |
| Criação do Cromossomo e População | Aleatória, mas com restrição para garantir o bom desempenho do AG |
| Nota Avaliação | Definida de acordo com as características compatíveis entre os alunos e os grupos definidos |
| Operadores Genéticos | |
| Seleção | Método Roleta |
| Cruzamento | <i>Crossover</i> de ponto único |
| Mutação | Inversão de Bit |

Definidos dados de entrada, restrições e modelagem do AG, teve início a implementação em módulos da ferramenta. Desenvolvido inicialmente o Algoritmo Genético conforme modelado, seguido da codificação dos módulos para entrada dos dados e, por fim, a implementação do módulo de resultados que permite a visualização dos grupos formados. A etapa final do desenvolvimento deste trabalho consistiu na integração dos módulos, para se ter uma ferramenta completa e funcional, pronta para ser utilizada em salas de aulas presenciais.

2. RESULTADOS E DISCUSSÃO

Os resultados da modelagem e desenvolvimento da ferramenta para agrupamento criada neste trabalho são expostos nesta seção, seguidos da apresentação de alguns testes realizados utilizando a ferramenta.

3.1 ENTRADA DE DADOS

O algoritmo desenvolvido neste trabalho tem como dados de entrada as informações relacionadas aos alunos e aos grupos. Estas informações são incluídas na base de dados por meio da *interface* desenvolvida que permite a interação do usuário com a ferramenta. Sendo necessário para cada aluno informar nome e até três características, conforme apresentado a seguir.

Figura 12 - Interface para cadastro das informações dos alunos



The image shows a user interface for student registration. It consists of two input fields. The first field is labeled "Aluno:" and contains the text "João". The second field is labeled "Característica Aluno:" and contains the text "HTML, CSS, AngularJS".

Fonte: Autoria própria

Aos grupos é necessário atribuir um nome identificador, as características desejadas e o limite de integrantes. A *interface* utilizada pelo usuário para entradas destes dados é apresentada na imagem abaixo.

Figura 13 - Interface para cadastro das informações dos grupos

Grupo:

Característica Grupo:

Limite de Integrantes:

Fonte: Autoria própria

Após a inserção dos dados necessários, o Algoritmo Genético realiza o processo de agrupamento baseado nos métodos de seleção natural, estes são descritos e exemplificados a seguir.

3.2 MODELAGEM E CODIFICAÇÃO DO PROBLEMA

Os dados informados pelo usuário através da *interface* apresentada nas imagens 12 e 13, são resgatados da base de dados padrão fornecida pelo *Framework Django* e guardados em listas auxiliares, utilizadas durante a execução do Algoritmo Genético (AG). Considerando que o usuário final realizou o cadastro dos dados de oito alunos e dois grupos, conforme apresentados na tabela a seguir.

Tabela 5 - Exemplo de informações cadastradas na ferramenta

| Dados Alunos | | Dados Grupos | | |
|--------------|----------------|--------------|----------------|--------------------|
| Nome | Característica | Nome | Característica | Limite Integrantes |
| Júlio | Exatas | Grupo 01 | Exatas | 04 |
| Lucas | Humanas | | | |
| Amanda | Exatas | | | |
| Paulo | Exatas | | | |

| | | | | |
|--------|------------|-------------|---------|----|
| Bruna | Exatas | Grupo 02 | Humanas | 04 |
| Carlos | Biológicas | | | |
| Ana | Humanas | | | |
| Júlia | Biológicas | | | |

As listas auxiliares utilizadas no AG recebem os dados dos alunos no seguinte formato:

Tabela 6 - Demonstração das listas auxiliares

| |
|---|
| nome_aluno['Julio', 'Lucas', 'Paulo', 'Amanda' ...] caracteristicas_aluno['Exatas', 'Humanas', 'Exatas', 'Exatas' ...] |
|---|

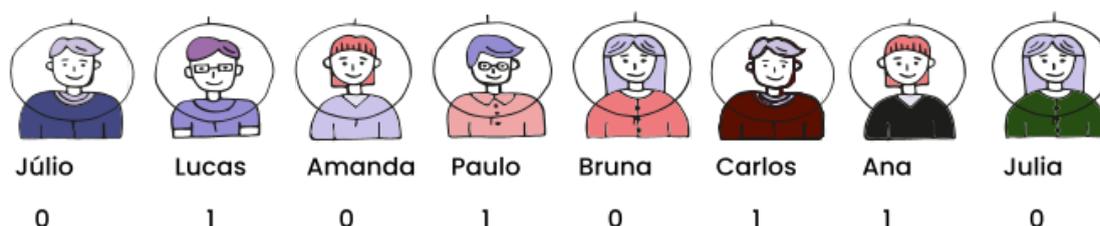
O mesmo se aplica para as informações do grupo, em que seus dados são salvos em listas similares às apresentadas acima.

A seguir será apresentado a estrutura de dados escolhida para codificar o problema. Desse modo, foi possível trabalhar de maneira computacional com o problema de agrupamento apresentado neste trabalho.

1. Codificação

Após coleta e armazenamento dos dados referente aos alunos e grupos, o AG realiza a codificação dos dados dos alunos de forma binária, ou seja, cada aluno passa a ser representado no algoritmo como um bit de valor igual a 0 ou 1.

Figura 14- Representação binária dos alunos



Conforme exemplo ilustrado acima, cada bit de valor binário corresponde a um aluno e, este conjunto de bits representa um indivíduo (cromossomo) no algoritmo, conforme representação a seguir.

Tabela 7 - Representação de um Indivíduo (cromossomo)

| | | | | | | | | |
|------------------|---|---|---|---|---|---|---|---|
| Indivíduo | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
|------------------|---|---|---|---|---|---|---|---|

Para ser possível a representação desses alunos no formato exposto acima, foi desenvolvida uma função para inicialização do indivíduo (cromossomo) e geração de uma população.

2. Inicialização do indivíduo e população

Os indivíduos (cromossomos) da população são inicializados de forma aleatória, baseando-se em um valor randômico, que define se o bit correspondente ao aluno terá o valor igual a 0, o que indica que este não fará parte da população, ou se receberá valor 1.

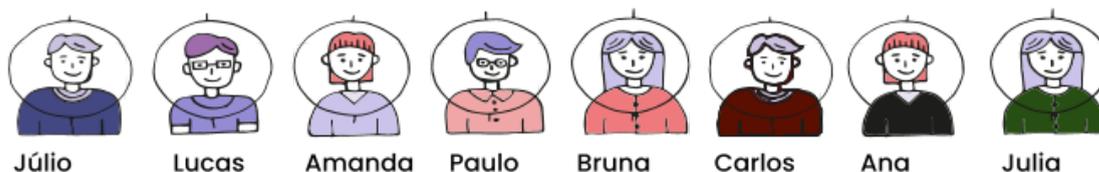
Para isto ser possível, a função do AG responsável por esta etapa se baseia no tamanho da lista auxiliar que contém o nome dos alunos, para gerar a nova lista preenchida com os valores binários, o que origina os indivíduos utilizados pelo AG no processo de avaliação, seleção, cruzamento e mutação. A seguir, no Quadro 1 é apresentado o pseudocódigo desta função.

Quadro 1 - Pseudocódigo Iniciar Indivíduo/População

| |
|---|
| <ol style="list-style-type: none"> 1. Recebe tamanho da lista auxiliar nome_alunos 2. Enquanto tamanho_lista_nome_aluno: <ul style="list-style-type: none"> Gera-se um valor aleatório Se valor > 0.5: <ul style="list-style-type: none"> indivíduo = '1' Caso contrário: <ul style="list-style-type: none"> indivíduo = '0' |
|---|

Após este procedimento é gerado um indivíduo com formato semelhante ao ilustrado a seguir.

Tabela 8 - Indivíduo gerado a partir de uma função aleatória

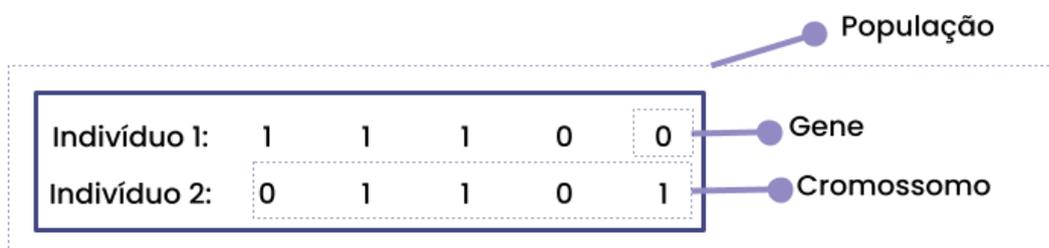


| | | | | | | | | |
|-------------------------|---|---|---|---|---|---|---|---|
| Indivíduo Gerado | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
|-------------------------|---|---|---|---|---|---|---|---|

Este processo é repetido até se ter um conjunto de novos indivíduos, que formam uma população diversificada e, que, por sua vez, tem o tamanho definido conforme a necessidade

do problema. A imagem 14 mostra a representação geral da estrutura do Algoritmo genético adotado no desenvolvimento deste trabalho.

Figura 15- Codificação Binária - Resultados



A seguir, é apresentada uma população fictícia para exemplificação, com tamanho estipulado a 100 indivíduos.

Tabela 9 - População - conjunto de indivíduos

| | | | | | | | | |
|----------------------|---|---|---|---|---|---|---|---|
| Indivíduo 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| Indivíduo 2 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| Indivíduo 3 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| Indivíduo 100 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |

Apesar da geração de indivíduos ocorrer de forma aleatória, foi necessário definir duas restrições para esta função, de modo a manter o bom funcionamento do AG, sendo:

- a quantidade de bits com valor 1 por indivíduo, não pode ser superior ao número limite de integrantes definidos para o grupo;
- um aluno já escolhido como integrante de um grupo, não pode ter seu bit com valor igual a 1, assim é possível evitar que um aluno faça parte de mais de um grupo.

Após a criação de uma população, é necessário avaliar cada um dos seus indivíduos e atribuir a estes uma nota de aptidão, conforme seu desempenho para ser uma boa solução ou não para o problema, neste caso, se o indivíduo contém o melhor conjunto de bits que representam os alunos certos para o grupo trabalhado no momento.

1. Nota e Avaliação dos indivíduos

A função de avaliação consiste em percorrer toda a população para verificação dos bits de cada um dos indivíduos. Caso o bit do indivíduo tenha valor igual a 1, o AG busca nas listas auxiliares os dados do aluno representado por este bit e verifica se a característica do

aluno corresponde à esperada para o grupo e, com base nesta informação, atribui uma nota ao indivíduo.

A seguir, será utilizado como exemplo, o Indivíduo 1 da população apresentado no tópico anterior, Tabela 9 - População - conjunto de indivíduos.

Tabela 10 - Funcionamento da Função Avaliação - Etapa 01

| | | | | | | | | |
|-----------------------------------|--|--|---|--|---|---|--|--|
| Indivíduo 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| Aluno Representado |  Júlio |  Lucas |  Amanda |  Paulo |  Bruna |  Carlos |  Ana |  Julia |
| Características Aluno | Exatas | - | | Exatas | | - | Humanas | - |
| Característica Do Grupo 01 | Exatas | | | | | | | |

Com todas as informações necessárias, a função de Avaliação atribui a nota do indivíduo da seguinte maneira: caso o aluno representado pelo bit com valor igual a 1 tenha a característica compatível com a esperada para o grupo, este recebe um acréscimo de 0,75 ao seu valor padrão que inicialmente é 0, caso contrário, terá sua nota reduzida em 0,30. Conforme ilustrado na tabela abaixo:

Tabela 11 - Funcionamento da Função Avaliação - Etapa 02

| | | | | | | | | |
|-----------------------------------|-------|---|---|-------|-------|---|--------|---|
| Indivíduo 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| Nota por Bit | +0,75 | 0 | 0 | +0,75 | +0,75 | 0 | - 0,30 | 0 |
| Nota do Indivíduo 1 = 1,95 | | | | | | | | |

Ao final, a nota do indivíduo é a soma da nota individual de cada bit.

1. Seleção

Com todos os indivíduos da população avaliados é o momento de selecionar os pais da próxima geração, ou seja, os responsáveis pelo processo de criar os indivíduos participantes da população seguinte.

A etapa de seleção dos pais utiliza o método Roleta, que simula a existência de uma circunferência (ou roleta) e cada indivíduo da população tem uma fatia desta roleta de acordo

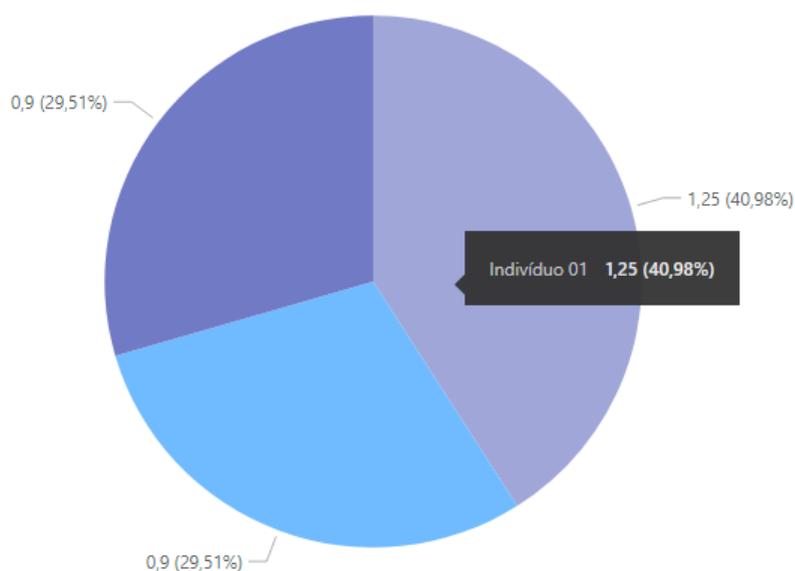
com sua nota. Em seguida, uma bolinha é lançada sobre esta circunferência para selecionar um indivíduo, conforme conceito apresentado na subseção 1.1.1.2, Seleção de Indivíduos para Próxima Geração. A seguir, é descrito o pseudocódigo para representação da função implementada.

Tabela 12 - Pseudocódigo Função Seleção

| |
|---|
| <ol style="list-style-type: none"> 1. Nota_população = soma nota de todos os indivíduos da população; 2. Gera um valor aleatório baseado na Nota_população; 3. Percorre população: <ol style="list-style-type: none"> a. Se indivíduo[i].nota_avaliação igual ou próximo a valor aleatório: Seleciona indivíduo[i] para ser Pai b. Caso contrário: continua a procurar um Pai |
|---|

Para representação do método roleta, é inicialmente somado o valor de aptidão de todos os indivíduos da população para gerar um valor base, este representa o todo da circunferência. Em seguida, cada indivíduo ocupa uma fatia da roleta, de acordo com sua nota de aptidão, conforme ilustrado abaixo.

Figura 16 - Exemplo do método roleta



Com todos os indivíduos ocupando seu espaço na roleta, a função de seleção gera um valor aleatório baseado na nota geral da população e, o indivíduo com nota igual ou mais

próxima ao número sorteado, é escolhido para ser o primeiro pai da nova população. O mesmo processo é realizado para seleção do segundo pai. Desse modo, pode ser sorteado um valor alto para seleção do melhor indivíduo(s) da população ou um valor discreto para seleção dos menos aptos.

1. Cruzamento e mutação

Após a seleção dos pais, é necessário realizar o cruzamento destes indivíduos, para se ter seus descendentes, estes farão parte da nova população. O método de cruzamento definido foi o de Ponto Único, o que permite que cada par de indivíduos pai gerem dois filhos, conforme o ponto de corte definido.

A seguir é apresentado o pseudocódigo da função responsável por este processo.

Tabela 13 - Pseudocódigo Função Cruzamento

| |
|--|
| 1. Recebe indivíduo_pai_01, indivíduo_pai_02 |
| 2. Ponto_corte = valor aleatório * tamanho indivíduo_pai |
| 3. Filho1 = indivíduo_pai_01 [0:Ponto_corte] + indivíduo_pai_02 [Ponto_corte::] |
| 4. Filho2 = indivíduo_pai_02 [0:Ponto_corte] + indivíduo_pai_01 [Ponto_corte::] |
| 5. Retorna Filho1, Filho2 |

A função de cruzamento desenvolvida define um Ponto Único de Corte baseado no tamanho dos indivíduos pais. Em seguida, o primeiro filho recebe a parte inicial do **indivíduo_pai_01** até o ponto de corte estipulado e, o segundo pai contribui com seu cromossomo a partir do ponto de corte. Cada par de indivíduos pai tem capacidade para gerar dois filhos. O segundo descendente é formado de maneira inversa. O processo descrito é exemplificado a seguir.

Tabela 14 - Formação dos descendentes - Crossover

| Ponto de corte definido: 3 | | | | | | | | |
|----------------------------|---|---|---|---|---|---|---|---|
| Indivíduo_Pai_01 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| Indivíduo_Pai_02 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| Filho 01 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| Filho 02 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |

Desse modo, a população seguinte é gerada de filhos, resultantes do processo de cruzamento dos indivíduos da população anterior.

O processo de mutação, necessário para gerar a evolução da população, consiste na inversão do valor de algum bit de um dos indivíduos da população recente. O processo de mutação é aplicado de maneira bem menos frequente, pois esta ação pode alterar a direção do algoritmo, tornando um bom indivíduo para o AG em uma solução menos interessante ou vice-versa.

Após estes processos, os novos e possivelmente melhorados indivíduos, resultantes do processo de seleção, cruzamento e mutação, são avaliados novamente, para que o AG verifique se a melhor solução possível foi encontrada, caso contrário o AG continua a criar gerações em busca de melhores soluções.

Todo o processo descrito para escolha dos integrantes do primeiro grupo é repetido para a formação do(s) grupo(s) seguinte(s), considerando a restrição de que o indivíduo já escolhido como solução final não deve estar presente no novo processo de seleção.

A seguir é apresentado o resultado obtido com a utilização da ferramenta.

4.2 RESULTADO DO MÉTODO PROPOSTO

Os testes realizados com intuito de avaliar o comportamento do Algoritmo Genético para formação de grupos conforme método proposto neste trabalho, considerou os seguintes parâmetros: a população do Algoritmo Genético foi composta por 100 indivíduos recombinados em busca da melhor solução por 150 gerações e, com taxa de mutação igual a 0,01.

Para utilização da ferramenta foi cadastrado na base de dados da ferramenta a informações fictícias de quatorze alunos, estes apresentados na tabela a seguir, juntamente com os nomes, características e limites de integrantes definidos para os grupos.

Tabela 15 - Configuração de grupos para testar o AG desenvolvido

| Dados Alunos | |
|------------------------------------|-------------------|
| Nome | Característica(s) |
| André, Júlia, Fabiano, Lucas | Python |
| Giovanna, Pedro, Carlos, Valdirene | UX |

| | |
|----------------------|---------------------------|
| Anna, Pablo, Carol | Flutter |
| Girlene, João, Bruno | JS, AngularJS, HTNL, CSS. |

| Dados Grupos | | |
|--------------|-------------------|--------------------|
| Nome | Característica(s) | Limite Integrantes |

| | | |
|----------|---------|----|
| BackEnd | Python | 04 |
| Designer | UX | 04 |
| Mobile | Flutter | 03 |

| | | |
|----------|--------------------------|----|
| FrontEnd | AngularJS, JS, HTML, CSS | 03 |
|----------|--------------------------|----|

Esta configuração inclui somente alunos com características iguais às esperadas para o grupo. O algoritmo, na configuração apresentada acima foi executado (testado) por várias vezes, sempre apresentando como saída o resultado conforme esperado. Ou seja, grupos compostos por alunos com características semelhantes entre si e, iguais às definidas para o grupo. O resultado descrito é apresentado na imagem a seguir.

Figura 17 - Resultado do teste do AG desenvolvido

GRUPO: BACKEND - ['FABIANO', 'JULIA', 'ANDRE', 'LUCAS']

GRUPO: DESIGNER - ['CARLOS', 'VALDIRENE', 'PEDRO', 'GIOVANNA']

GRUPO: FRONTEND - ['BRUNO', 'JOÃO', 'GIRLENE']

GRUPO: MOBILE - ['CAROL', 'PABLO', 'ANNA']

Fonte: Autoria própria

Contudo, uma sala de aula comporta diversos perfis de alunos, sendo assim espera-se a existência de alunos com características divergentes às propostas para o(s) grupo (s) e que não devem ser desconsiderados durante este processo. Desta forma, para representar o cenário descrito, a característica de dois alunos foi alterada de forma que os tornaram incompatíveis com os grupos a serem criados. Conforme apresentado na tabela abaixo.

Tabela 16 - Configuração de grupos para testar o AG desenvolvido - Alunos Divergentes

| Dados Alunos | |
|------------------------------------|---------------------------------|
| Nome | Característica(s) |
| Lucas | MySql, Oracle, BD |
| André, Júlia, Fabiano | Python |
| Giovanna, Pedro, Carlos, Valdirene | UX, UI, Motion, Designer |

| | |
|----------------------|---------------------------|
| Anna, Pablo, Carol | Flutter |
| João, Girlene, Bruno | HTML, CSS, JS, AngularJS. |

| Dados Grupos | | |
|---------------------|--------------------------|---------------------------|
| Nome | Característica(s) | Limite Integrantes |

| | | |
|----------|---------|----|
| BackEnd | Python | 04 |
| Designer | UX | 04 |
| Mobille | Flutter | 03 |

| | | |
|----------|-----------------------------|----|
| FrontEnd | AngularJS, JS, HTML, CSS | 03 |
|----------|-----------------------------|----|

Com esta configuração o AG apresenta resultados menos favoráveis. Conforme os resultados frequentes apresentados pelo AG os grupos iniciais são montados com a maioria dos alunos que possuem características compatíveis. A imagem 18 apresenta o resultado exibido após algumas execuções do algoritmo, considerando a configuração de grupo apresentada na Tabela 16 para testar o Algoritmo desenvolvido.

Figura 18 - Resultado do teste do AG desenvolvido — Alunos Divergentes

GRUPO: BACKEND - ['GIRLENE', 'FABIANO', 'JULIA', 'ANDRE']

GRUPO: DESIGNER - ['CAROL', 'CARLOS', 'PEDRO', 'GIOVANNA']

GRUPO: FRONTEND - ['BRUNO', 'JOÃO', 'PABLO']

GRUPO: MOBILE - ['ANNA', 'VALDIRENE', 'LUCAS']

Fonte: Autoria própria

O resultado mostra que os acadêmicos que não se enquadram no perfil esperado para o grupo são desconsiderados pelo algoritmo e integrados ao último grupo. Este resultado é consequência da maneira como a nota de aptidão é atribuída a cada indivíduo, conforme compatibilidade entre características do aluno com o grupo, sempre que um indivíduo é estruturado com um ou ambos alunos com características inesperadas, este durante o processo de avaliação é penalizado, tendo sua nota de aptidão reduzida, e assim, se torna um indivíduo menos interessante para solução final e como consequência são desconsiderados como parte dos grupos iniciais e como solução o AG os associa ao último grupo.

Desse modo, o AG quando utilizado para formação de grupos heterogêneos conforme representados na Tabela 16, tende a apresentar resultados menos favoráveis.

3. CONSIDERAÇÕES FINAIS

O agrupamento adequado dos alunos desempenha um papel importante relacionado a apoiar o desenvolvimento de habilidades individuais e coletivas dos alunos, bem como da atividade proposta.

Neste sentido, o algoritmo genético trabalhado permitiu a exploração combinatória eficiente de um conjunto de características dos alunos e direcionada conforme as restrições definidas para os grupos. A partir do desenvolvimento deste trabalho e das análises realizadas é possível identificar que a ferramenta criada obteve os melhores resultados quando utilizada para formação de grupos homogêneos, com abordagem selecionada, seguindo os critérios definidos pelo professor.

Contudo, por ter como base somente as características dos alunos e grupos para realizar o processo de agrupamento, a ferramenta apresenta dificuldade para encaixar estudantes com perfis diferentes. Neste contexto, para trabalhos futuros é necessário organizar uma nova forma de avaliar e selecionar os alunos ou realizar melhorias a parte do processo evolutivo do Algoritmo Genético pertinente ao processo de avaliação, que até o momento utiliza da compatibilidade das características do aluno com o grupo para atribuição da nota de aptidão, que influencia consideravelmente na oportunidade de um aluno ser ou não selecionado como parte da solução final.

A trabalhos futuros que proponham o desenvolvimento de uma nova ferramenta, sugere-se, na etapa de entendimento do problema, o estudo e análise da possibilidade de se utilizar novas variáveis que possam ser usadas como parâmetros para o algoritmo gerar grupos mais eficientes.

4. REFERÊNCIAS

BALIEIRO, Andson et al. Uma Abordagem Baseada em Algoritmo Genético para Formação de Grupos de Estudos em Ambientes Virtuais de Aprendizagem. **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)**, [S.l.], p. 1287, out. 2017. ISSN 2316-6533. Disponível em: <<https://www.br-ie.org/pub/index.php/sbie/article/view/7657>>. Acesso em: 22 ago. 2020.

CAETANO, Samuel Sabino. **O uso de algoritmos evolutivos para a formação de grupos na aprendizagem colaborativa no contexto corporativo**. 2013. 149 f. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal de Goiás, Goiânia. 2013. Disponível em: <<https://repositorio.bc.ufg.br/tede/handle/tede/3195>>. Acesso em: 29 nov. 2020.

CASTRO, Leandro et al.. **Computação Natural: Uma Breve Visão Geral**. 2004. Researchgate. Disponível em: <<https://www.researchgate.net/publication/261438133>>. Acesso em: 10 de set. 2020

EIBEN, A.E; SCHOENAUE, M. Evolutionary computing. **Science Direct**, [S.I.], v. 82, n. 1, p. 1-6, 15 abr. 2002. Disponível em: <[https://doi.org/10.1016/S0020-0190\(02\)00204-1](https://doi.org/10.1016/S0020-0190(02)00204-1)>. Acesso em: 14 set. 2020.

GENÇER, Hüseyin. Group Dynamics and Behaviour. **Universal Journal Of Educational Research**: ERIC. Turkey, p. 223-229. 2019. Disponível em: <<https://eric.ed.gov/?id=EJ1201477>>. Acesso em: 08 set. 2020.

GOEDERT, Matheus L.; PAULA FILHO, Pedro L.; BLANCO, Daniel R. Computação natural:: conceitos e aplicações da computação inspirada na natureza. **Revista Espacios**, [S. L.], v. 38, n. 34, p. 31-31, mar. 2017. Disponível em: <<https://www.revistaespacios.com/a17v38n34/a17v38n34p31.pdf>>. Acesso em: 10 set. 2020

GROUP Work in the Classroom: Definition, Benefits & Examples. 2017. **Study.com**. Disponível em: <<https://study.com/academy/lesson/group-work-in-the-classroom-definition-benefits-examples.html>>. Acesso em: 06 set. 2020.

HOLLAND, John H.. Genetic Algorithms. **Scientific American**: a division of Nature America, Nova Iorque, v. 267, n. 1, p. 66-73, jul. 1992. Disponível em: <<https://www.jstor.org/stable/24939139>. Acesso em: 06 set. 2020>.

IEEE ([S.I]). Computational Intelligence Society. **What is Computational Intelligence?** [20--?]. Disponível em: <<https://cis.ieee.org/about/what-is-ci>>. Acesso em: 10 set. 2020.

IYODA, Eduardo Masato. . **Inteligência computacional no projeto automático de redes neurais híbridas e redes neurofuzzy heterogêneas**. 2000. 166 f. Dissertação (Mestrado) - Curso de Engenharia Elétrica, Departamento de Engenharia de Computação e Automação Industrial, Universidade Estadual de Campinas, São Paulo, 2000. Disponível em: <<http://repositorio.unicamp.br/jspui/handle/REPOSIP/259071>>. Acesso em: 10 set. 2020.

JOHNSON, Roger T.; JOHNSON, David W.; SMITH, Karl A.. Active Learning: Cooperation in the Classroom. **The Annual Report Of Educational Psychology In Japan**. [Tóquio], p. 29-30. jan. 1998. Disponível em: <https://www.researchgate.net/publication/234568124_Active_Learning_Cooperation_in_the_College_Classroom>. Acesso em: 20 nov. 2020.

JOHNSON, David W.; JOHNSON, Roger T.. An Educational Psychology Success Story:: social interdependence theory and cooperative learning. **Sage Journals**. [S.I], p. 365-379. jun. 2009. Disponível em: <<https://journals.sagepub.com/doi/10.3102/0013189X09339057>>. Acesso em: 20 nov. 2020.

LINDEN, Ricardo. GAs: conceitos básicos. In: LINDEN, Ricardo. **Algoritmos Genéticos**: uma importante ferramenta da inteligência computacional. 2. ed. Rio de Janeiro: Brasport, 2008. Cap. 1. p. 40-48. Disponível em: <<https://books.google.com.br/books?id=it0kv6UsEMEC&printsec=frontcover&hl=pt-BR#v=onepage&q&f=false>>. Acesso em: 06 set. 2020.

MARZANO, Robert J.; PICKERING, Debra J.; POLLOCK, Jane E.. Cooperative Learning. In: MARZANO, Robert J.; PICKERING, Debra J.; POLLOCK, Jane E.. **Classroom**

Instruction That Work: research - based strategies for increasing student achievement. Alexandria Virgínia: Ascd, 2004. Cap. 7. p. 84-88. Disponível em: <<https://books.google.com.br/books?id=c25kDO0adxwC&printsec=frontcover&hl=pt-BR#v=onepage&q&f=false>>. Acesso em: 21 nov. 2020.

MEDEIROS, Luciano Frontino de. Introdução a algoritmos genéticos. In: MEDEIROS, Luciano Frontino de. **Inteligência Artificial Aplicada:** uma abordagem introdutória. Curitiba: Editora Intersaberes, 2018. Cap. 6. p. 165-179. Disponível em: <<https://plataforma.bvirtual.com.br/Acervo/Publicacao/161682>>. Acesso em: 08 set. 2020

MELLO, Clivanei Gonçalves de; ARIMA, Carlos; NEVES, José Manoel Souza das. Formação e características de times, equipes ou grupos e características dos membros para atuar em sistemas dinâmicos. In: IX WORKSHOP DE PÓS-GRADUAÇÃO E PESQUISA DO CENTRO PAULA SOUZA, 9., 2014, São Paulo. **Anais [...]** . São Paulo: Cps, 2013. p. 439-448. Disponível em: <<http://www.pos.cps.sp.gov.br/files/artigo/file/482/81986a69b9c09006e8d52f7af740b21b.pdf>>. Acesso em: 29 nov. 2020.

MIRANDA, Péricles *et al.* Formação de grupos de alunos baseado em múltiplos critérios. **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)**, [S.l.], p. 1607, out. 2017. ISSN 2316-6533. Disponível em: <<https://www.br-ie.org/pub/index.php/sbie/article/view/7692>>. Acesso em: 08 set. 2020.

MORENO, Julián; OVALLE, Demetrio A.; VICARI, Rosa M. A genetic algorithm approach for group formation in collaborative learning considering multiple student characteristics. **ScienceDirect**, [S.l.], v. 58, n. 1, p. 560-569, jan. 2012. Disponível em: <<https://doi.org/10.1016/j.compedu.2011.09.011>>. Acesso em: 31 ago. 2020.

OBITKO, Marek. **Introdução aos Algoritmos Genéticos.** 1998. Disponível em: <<https://www.obitko.com/tutorials/genetic-algorithms/portuguese/index.php>>. Acesso em: 18 set. 2020.

OUNNAS, Asma. **Enhancing the Automation of Forming Groups for Education with Semantics.** 2010. 190 f. Tese (Doutorado) - Curso de Faculty Of Engineering And Applied

Science, Department Of Electronics And Computer Science, University Of Southampton, Southampton, 2010. Disponível em: <<https://eprints.soton.ac.uk/171641/>>. Acesso em: 30 ago. 2020.

OUNNAS, Asma; DAVIS, Hugh C; MILLARD, David E. Towards Semantic Group Formation. **Seventh Ieee International Conference On Advanced Learning Technologies: ICALT 2007**, Niigata, p. 828-827, 2007. Disponível em: <<https://ieeexplore.ieee.org/document/4281168>>. Acesso em: 08 set. 2020.

PILA, Adriano Donizete. História e terminologia a respeito da computação evolutiva. **Kroton: Revista de Ciências Exatas e Tecnologia**, [S. L.], v. 1, n. 1, p. 40-50, 13 jul. 2015. Disponível em: <<https://doi.org/10.17921/1890-1793.2006v1n1p40-50>>. Acesso em: 13 set. 2020.

SRBA, Ivan; BIELIKOVA, Maria. Dynamic Group Formation as an Approach to Collaborative Learning Support. **Ieee Transactions On Learning Technologies**. [S.I.], p. 173-186. jun. 2015. Disponível em: <<https://www.computer.org/csdl/journal/lt/2015/02/06963449/13rRUwInvoy>>. Acesso em: 21 nov. 2020.

TSURUTA, Jaime Hidehiko; NARCISO, Marcelo Gonçalves. Um estudo sobre algoritmos genéticos. **Embrapa Informática Agropecuária: Documentos (INFOTECA-E)**. Campinas. maio 2001. Disponível em:<<https://www.infoteca.cnptia.embrapa.br/infoteca/handle/item/291>>. Acesso em: 25 jan. 2021

URSEM, Rasmus K.. Diversity-Guided Evolutionary Algorithms. In: PARALLEL PROBLEM SOLVING FROM NATURE — PPSN VII, 8., 2002, Granada. **Artigo de conferência**. Granada: Springer, Berlim, Heidelberg, 2002. v. 2439, p. 462-471. Disponível em:< https://doi.org/10.1007/3-540-45712-7_45>. Acesso em: 24 nov. 2020.

ZHAMRI, Che; ANI, Zhamri Che; YASIN, Azman; HUSIN, Mohd Zabidin; HAMID, Zauridah Abdul. A Method for Group Formation Using Genetic Algorithm. **(Ijcs) International Journal Of Advanced Trends In Computer Science And Engineering**. [S. L.], p. 3060-3064. jan. 2010. Disponível em:

<https://www.researchgate.net/publication/229035954_A_Method_for_Group_Formation_Using_Genetic_Algorithm>. Acesso em: 24 ago. 2020.

ZUBEN, Fernando J. Von. Computação evolutiva: uma abordagem pragmática. **Tutorial: Notas de Aula da disciplina IA707, Faculdade de Engenharia Elétrica e de Computação-Universidade Estadual de Campinas**, 2000. Disponível em: <<https://www.ic.unicamp.br/~rocha/teaching/2011s1/mc906/aulas/computacao-evolutiva-uma-abordagem-pragmatica.pdf>>. Acesso em: 13 set. 2020