Revista **Ciências Exatas**
V.29, N°2, 2023 | Taubaté/SP - Brasil | **ISSN:** 1516-2893

FORMAÇÃO DE EQUIPES HOMOGÊNEAS COM O USO DE ALGORITMO GENÉTICO

FORMATION OF HOMOGENEOUS TEAMS USING GENETIC ALGORITHM

Chrystian Souza | Faculdade de Tecnologia de Cruzeiro

Lucas Mallmann | Faculdade de Tecnologia de Cruzeiro

Luis Fernando de Almeida | Faculdade de Tecnologia de Cruzeiro e UNITAU

Francisco José Grandinetti | UNESP/Guará

Alvaro Manoel Sousa Soares | UNESP/Guará

RESUMO

Trabalhar em equipe é uma das principais atividades que a humanidade realiza desde muito tempo. Nos últimos anos a literatura tem dado um cuidado especial ao tema formação de equipes, muito se discorre sobre isso, porém ainda existe uma grande dificuldade em se formar uma equipe homogênea, de forma a otimizar a produtividade. Baseado nesse contexto foi desenvolvido um algoritmo genético (AG) capaz de dividir e otimizar indivíduos em equipes baseando-se em suas habilidades individuais. Os testes realizados em diferentes configurações se mostraram bastante promissores, em alguns casos houve uma otimização de aproximadamente 95% em relação a primeira geração.

Palavras-chave: Equipe Homogênea. Algoritmo Genético. Otimização. Habilidades Individuais.

ABSTRACT

Working in a team is one of the main activities that mankind has been doing for a long time. In recent years the literature has given a special care to the topic of team formation, much is discussed about this, but there is still a great difficulty in forming a homogeneous team, to optimize productivity. Based on this context, a genetic algorithm (GA) capable of dividing and optimizing individuals into teams was developed based on their individual abilities. The tests performed in different configurations were very promising, in some cases there was an optimization of approximately 95% in relation to the first generation.

Keywords: Homogeneous team. Genetic Algorithm. Optimization. Individual Skills.

1 INTRODUÇÃO

Quando surge a necessidade de se formar uma equipe para realizar alguma atividade/projeto nota-se, em muitas situações, certa dificuldade quanto a escolha dos indivíduos. Em alguns casos, é comum observar que fatores, como afinidade com o recrutador se mostram mais relevantes do que as habilidades técnicas/interpessoais em si.

Já em outros utiliza-se a aleatoriedade ou até mesmo casualidade, para garantir uma divisão justa e imparcial (método muito comum, geralmente utilizado em salas de aula). No entanto, esta opção pode não ser uma boa ferramenta, visto que assim podemos obter grupos fortes demais e outros muito fracos, pois indivíduos com habilidades muito parecidas podem ficar no mesmo grupo, enquanto existirão grupos sem tais habilidades.

A literatura apresenta algumas pesquisas para esse problema. Por exemplo, Silva (2017) propõe uma solução utilizando Algoritmos Genéticos (AGs) para a formação de equipes heterogêneas em turmas universitárias. Segundo ele, o uso de AGs possibilitaria que a formação das equipes levasse em conta uma série de fatores determinantes para que a divisão dos alunos favoreça o aprendizado entre os indivíduos.

Silveira (2006), apresenta uma abordagem muito semelhante utilizando-se de AGs para formação de grupos colaborativos que irão trabalhar à distância, via *web*. Para tal, sua solução conta com critérios que poderão ser configurados pelo professor.

Destaca-se, também, o trabalho de Strnad e Guid (2010) que combinam as habilidades necessárias à equipe com as habilidades pessoais de cada indivíduo utilizando AGs. Para tal, optam por utilizar o conceito de lógica fuzzy, que ao contrário da lógica booleana que somente admite valor “verdadeiro” ou “falso” (0 ou 1), permite trabalhar com qualquer valor entre 0 e 1, que podem representar como sendo “quase verdade” ou “quase falso”, respectivamente.

Wi (2009) desenvolveu um método quantitativo e sistemático, no qual propõe uma estrutura para analisar o conhecimento dos candidatos a gerentes e membros da equipe e, também, um AG e medidas de redes sociais para a escolha de um gerente de equipe e membros da equipe.

Uma das maiores utopias das empresas está em criar uma única equipe de sucesso. Equipes têm o potencial de aumentar a produtividade baseada na reunião de talentos. Porém, apesar de todas as potencialidades que uma equipe oferece a literatura sugere que existe uma série de barreiras desde a formação ao amadurecimento que um grupo de pessoas deve superar para ser considerada uma equipe

(Bejarano; Pilatti; Lima, 2005).

Visto que a formação da equipe é um dos momentos importantes do projeto e pode ser transformar em uma etapa complicada, devido a infinidade de variáveis envolvidas, este presente trabalho propõe a implementação de uma solução algorítmica utilizando-se de AGs para formação de equipes homogêneas entre si, de forma que não seja possível notar diferença entre uma equipe e outra, com o propósito de que qualquer equipe formada tenha plena capacidade de resolver o problema proposto. Para o presente trabalho, foram utilizados três parâmetros de entrada apenas, referentes a pontos de habilidades atribuídos a cada membro da equipe.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 ALGORITMOS GENÉTICOS

Os AGs foram primeiramente propostos por Holland em dois de seus principais trabalhos (Holland, 1975; Holland, 1992). São algoritmos de otimização e busca baseados nos conceitos de seleção e genética natural propostos por Darwin (Holland, 1989). Ele é uma abstração da seleção natural e reprodução genética, fazendo com que os mais aptos se sobressaiam dos indivíduos menos aptos, possuindo maior probabilidade de que seu código genético seja passado para um número maior de gerações, resultando em populações cada vez mais aptas.

Por ser um algoritmo baseado na biologia, cada componente possui a sua representação computacional. Cada um dos indivíduos codificados na população pode ser visto como uma representação, de acordo com uma codificação apropriada de uma solução particular problema, conforme ilustrado no Quadro 1.

Quadro 1 - Analogia entre sistema natural e algoritmo genético.

Natureza	Algoritmo Genético
Cromossomo	Palavra binária, valor etc.
Gene	Característica do problema
Alelo	Valor da característica
Genótipo	Estrutura
Fenótipo	Estrutura submetida ao problema
Indivíduo	Solução
Geração	Ciclo

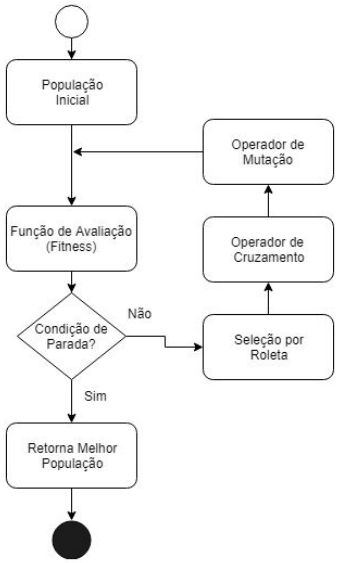
Os AGs são utilizados em situações em que se deseja saber qual é a melhor solução em um conjunto de soluções específico (SIVANANDAM; DEEPA, 2008). Para encontrar a melhor solução, o algoritmo realiza operações genéticas, como seleção, cruzamento e mutação da população, avaliando cada nova população para que ela esteja convergindo para o objetivo final.

A seleção é feita avaliando os indivíduos da população. Os indivíduos são selecionados de acordo com a avaliação de cada um de seus cromossomos, que contém um valor de avaliação da solução que representa. Primeiramente, é necessário gerar uma população inicial, que geralmente é gerada de forma aleatória, e deve conter o máximo de material genético divergente possível.

Após a população inicial ser gerada, um AG entra em ação para realizar o processo de evolução. A evolução ocorre através dos seguintes passos: seleção, cruzamento, avaliação e substituição. A seleção tem como objetivo selecionar indivíduos para a reprodução. No início, isso é feito de maneira aleatória com uma probabilidade dependendo da avaliação dos cromossomos, fazendo com que os melhores indivíduos tenham mais chance de se reproduzir.

A reprodução envolve a criação de novos indivíduos utilizando o cruzamento e a mutação. A avaliação é o processo responsável por analisar os indivíduos. A substituição é o último passo, responsável por substituir os antigos indivíduos por indivíduos mais novos e aptos na população. Assim sendo, o algoritmo continua neste ciclo, também chamado de geração, até que o critério de parada estabelecido tenha sido alcançado. A Figura 1 ilustra o fluxo básico de um AG.

Figura 1 | Funcionamento de um AG.



1.2 FORMAÇÃO DE EQUIPES

Muito se discute sobre a definição de "equipe" no ambiente empresarial. Alguns autores discorrem sobre como as empresas acreditam estarem formando equipes, quando na verdade estão apenas agrupando pessoas, como nota Drucker (2001).

Na verdade, uma equipe se caracteriza por ser uma união de pessoas, com habilidades complementares que se mantêm unidas com foco em um mesmo objetivo, que apesar de realizarem o trabalho independentemente são todos responsáveis pelos resultados (KATZENBACH; SMITH, 2001).

Uma equipe de alta desempenho está sempre motivada por um sentimento de conquista, de poder, pensar e agir para alcançar o objetivo comum a todos (BLANCHARD, 2000).

A maior parte das empresas está longe de alcançar esta definição de equipe, já que na maioria das vezes são apenas grupos de trabalho que executam suas atividades isoladamente e que estão longe de vislumbrar um objetivo comum. Como discorrem Larson e LaFasto (1989), a maior dificuldade de se formar uma equipe está no conflito entre os objetivos individuais e os objetivos do grupo.

Para muitos pesquisadores o ponto chave do sucesso de uma equipe está no exato momento da seleção. Segundo Larson e LaFasto (1989), é de extrema importância que a seleção seja baseada em dois níveis de capacitação: habilidades técnicas e habilidades interpessoais. Pois o mínimo que se espera de um candidato é que ela contenha capacidades técnicas que somadas as habilidades do grupo como um todo sejam complementares. Também é de vital importância que o indivíduo se sinta responsável pelo sucesso do grupo, seja capaz de lidar com pessoas e possua sempre o desejo de contribuir.

Sendo assim, formar equipes de alto desempenho, depende da combinação ideal das habilidades entre os membros que as compõem, de forma que não existam equipes melhores ou piores.

3. METODOLOGIA

3.1 FERRAMENTAS UTILIZADAS

Para desenvolvimento do projeto proposto foram utilizadas as ferramentas que serão, brevemente, apresentadas nos parágrafos subsequentes.

O Python 3.6 foi escolhido devido a sua habilidade notável em trabalhar com números e sua sintaxe simples, permitindo assim que o desenvolvedor possa focar convenções da linguagem. Python é

uma linguagem de programação que permite trabalhar rapidamente e integrar sistemas de forma mais eficaz (Python.org, 2018). O NumPy, abreviação de Python Numérico (em inglês, *Numerical Python*) é uma biblioteca que possui várias funções matemáticas, e se sua adoção se tornou popular nos últimos anos em lugares como: a academia, laboratórios nacionais e indústria, com aplicação abrangendo jogos computacionais até exploração espacial (Walt; Colbert; Varoquaux; 2011).

O Visual Studio Code é um editor de código *open source* desenvolvido pela empresa *Microsoft Corporation*. A ferramenta oferece suporte para diversas linguagens, facilitando assim o desenvolvimento. Ainda oferece suporte para *plugins* externos, permitindo assim ser personalizado conforme as necessidades do usuário.

3.2 SOLUÇÃO PROPOSTA

Para solucionar a questão da dificuldade no momento de se formar uma equipe, propõe-se um AG que será responsável por analisar as habilidades dos indivíduos envolvidos e separá-los em grupos, de forma que as equipes sejam o mais homogêneas possível.

3.1.1 Modelagem do problema

Como mencionado anteriormente, a formação da equipe é principal etapa do projeto, visto que os membros que a compõem serão responsáveis pelo desenvolvimento e conclusão dele. No entanto, ainda se tem certa dificuldade por parte dos gestores de se separar pessoas com habilidades distintas em grupos, muitas das vezes deixa-se levar em consideração variáveis como “afinidade” e “achismo”, o que já se provou não ser eficiente.

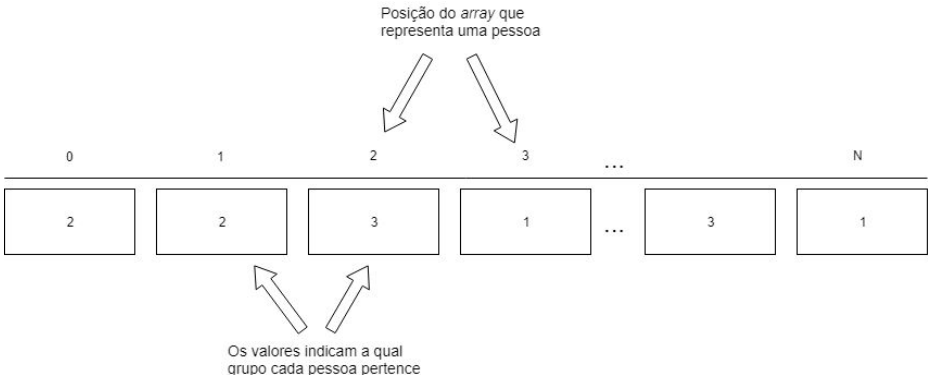
3.1.2 Modelagem do AG

Quando se projeta um AG torna-se necessário a definição de alguns componentes essenciais que irão ser fundamentais para a convergência dos resultados. São eles: cromossomo, população inicial, função de avaliação, operadores de cruzamento e de mutação e parâmetros genéticos. As subseções seguintes descrevem a configuração utilizada para cada um destes itens.

3.1.2.1 Estrutura do cromossomo

A estrutura do cromossomo se dá pela formação de um *array* de tamanho N, onde cada posição N representa o identificador de uma pessoa. O valor de cada posição indica a qual grupo a pessoa pertence. A Figura 2 ilustra um exemplo de cromossomo utilizado, considerando 4 indivíduos e 2 grupos.

Figura 2 | Exemplo de um cromossomo.



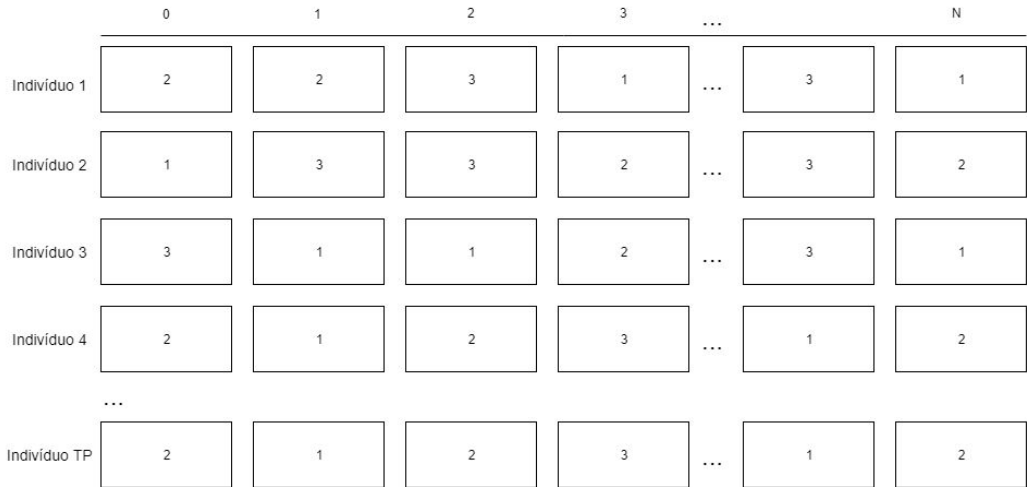
3.1.2.2 População inicial

Para iniciar o programa, gera-se uma matriz com um tamanho fixo pré-definido, que é chamado de tamanho da população (TP). A matriz é preenchida com TP indivíduos (cromossomos) criados de forma aleatória para formar a população inicial. A Figura 3 representa o algoritmo utilizado para gerar a população inicial e a Figura 4 ilustra um exemplo de população inicial, com um total de N pessoas divididas em 3 grupos.

Figura 3 | Algoritmo para geração da população inicial.

```
função PopulaçãoInicial(tamanho_população, tamanho_individuo) saídas: População
    repetir
        individuo := criar_individuo(tamanho_individuo)
        população.adicionar_indivíduo(indivíduo)
    enquanto
        tamanho_população não foi atingido
    retorna População
```

Figura 4 | Exemplo de população inicial.



3.1.2.3 Função de aptidão

O conceito utilizado de equipes homogêneas se dá do princípio de que todas as equipes possuem a mesma capacidade de resolver problemas, dados seus integrantes, dentro de um mesmo contexto. Assim sendo, ao se comparar equipes homogêneas, a diferença das habilidades entre elas deve tender a zero. Para tal, a função de avaliação utilizada tem como objetivo minimização.

Logo, ela se baseia na diferença das habilidades de cada equipe, de forma que quanto mais próximo de zero, melhor é a avaliação da equipe em questão. Sabendo a quantidade de grupos (qg), quantidade de parâmetros ou habilidade distintas (qp) e a quantidade de alunos (qa), a matriz de parâmetros é expressa da seguinte forma:

$$P_{qp,qa} = \begin{bmatrix} 1 & \dots & qa \\ \dots & \dots & \dots \\ qp & \dots & qa \end{bmatrix}$$

$$I_c = \{v_1,\dots,v_{qa}\} \text{ onde } c = 1\dots TP$$

A função de aptidão é dada por:

$$fit_c = \frac{1}{\sum_{i=1}^{qp} sum_i * sum_i}$$

$$sum_i = \sum_{j=1}^{qg-1} \sum_{jj=j+1}^{qg} |SPG_{I,J} - SPG_{I,JJ}| \text{ , para i = 1...qp}$$

$$SPG_{i,j} = \sum_{k=1}^{qa} b * P_{i,k} \ com \begin{cases} b = 0, \ se\ j \neq I_k \\ b = 1, \ caso\ contrário \end{cases}$$

para i=1...qp e j=1...qg

3.1.2.4 Operador de cruzamento

O cruzamento é realizado entre dois cromossomos, em que cada cruzamento gera 2 novos filhos. Para realizar o cruzamento é definido uma posição de corte, que irá "cortar" os cromossomos para combinar as partes e gerar os dois novos filhos. A Figura 5 demonstra como o processo de cruzamento acontece e a Figura 6 ilustra como os cromossomos se comportam durante o operador de cruzamento.

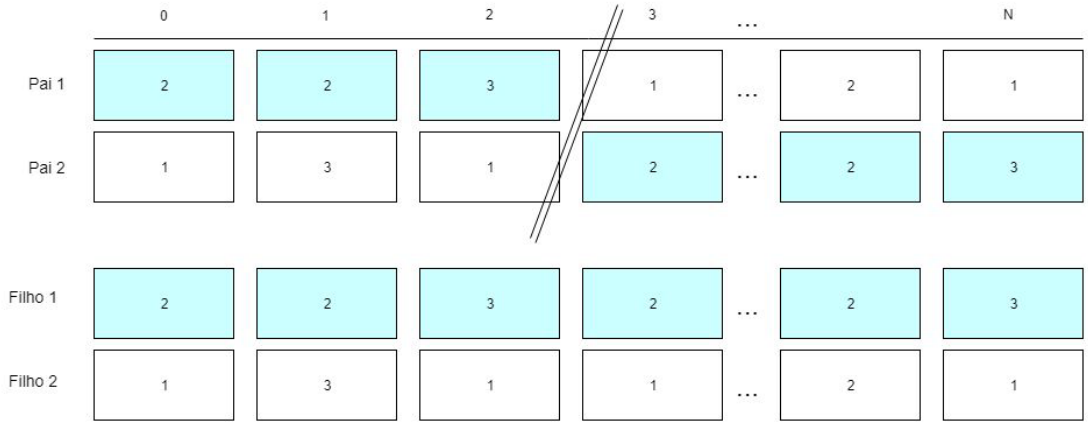
Figura 5 | Algoritmo para realizar o cruzamento.

```
função Cruzamento (posicao_corte) saídas: Filho1, Filho2
    pai1 = seleciona_por_roleta()
    pai2 = seleciona_por_roleta()

    Filho1 = pai1.cortar(posicao_corte) + pai2.cortar(posicao_corte)
    Filho2 = pai2.cortar(posicao_corte) + pai1.cortar(posicao_corte)

    retornar Filho1, Filho2
```

Figura 6 | Exemplo de cruzamento.

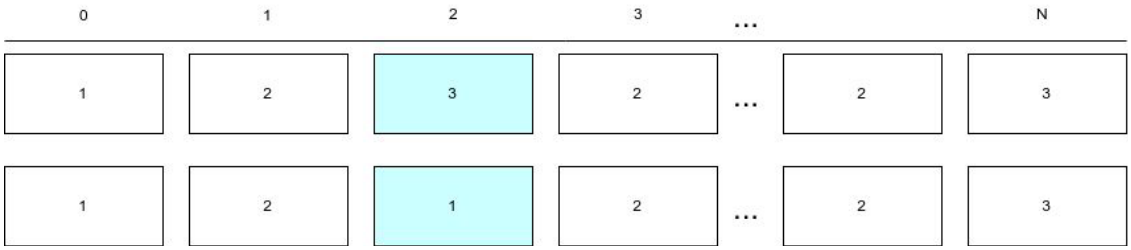


Após o cruzamento ser realizado, os filhos sofrem uma "correção", para que a distribuição das pessoas seja feita de forma uniforme. Essa correção irá distribuir as pessoas igualmente entre os grupos, para que não haja nenhum grupo com mais ou menos membros. O algoritmo para esta correção está descrito na Figura 7 e a Figura 8 ilustra a situação a qual foi necessário que a pessoa 2 saísse do grupo 3 para o grupo 1, pois a divisão das pessoas por grupo não estava correta.

Figura 7 | Algoritmo para correção do cromossomo após cruzamento.

```
função CorrigirCromossomo(cromossomo) retorna Cromossomo
    para cada numero_do_grupo em grupos
        contador := cromossomo.contar(numero_do_grupo)
        enquanto contador != pessoas_por_grupo
            mudar_pessoa_de_grupo(cromossomo)
    retorna cromossomo
```

Figura 8 | Ajuste do cruzamento.



3.1.2.5 Operador de mutação

A mutação ocorre nos cromossomos, através da translocação. Define-se uma parcela dos descendentes da população que sofrerá mutação, em sequência os cromossomos são selecionados aleatoriamente para serem mutados. Após selecionados, um indivíduo do cromossomo irá trocar de posição com outro (translocado), aleatoriamente dentro do mesmo cromossomo. Este processo ocorrerá em cada cromossomo selecionado para a mutação. A Figura 9 descreve o algoritmo utilizado para realizar a mutação e a Figura 10 ilustra um exemplo da mutação.

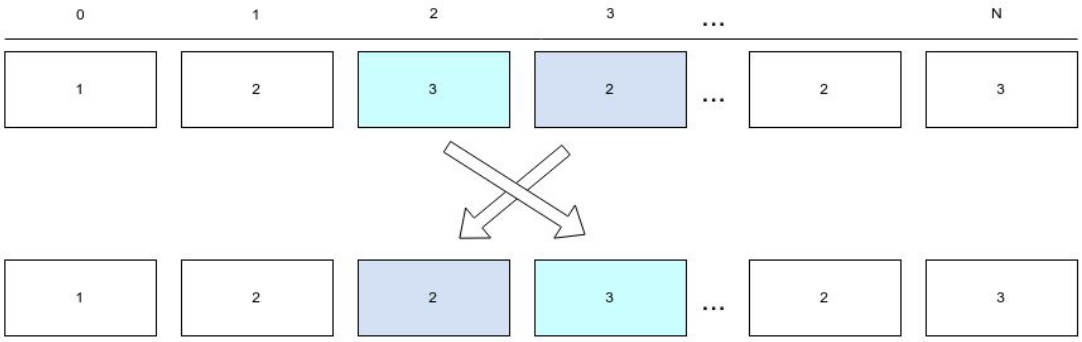
Figura 9 | Algoritmo para realizar a mutação.

```
função mutacao(cromossomo) retorna Cromossomo
    posicao1 = cromossomo.posicao_aleatoria()
    posicao2 = cromossomo.posicao_aleatoria()

    se posicao1 != posicao2
        auxiliar = cromossomo[posicao1]
        cromossomo[posicao1] = cromossomo[posicao2]
        cromossomo[posicao2] = auxiliar

    retorna cromossomo
```

Figura 10 | Mutação.



3.1.2.6 Parâmetros genéticos

Para criação do algoritmo foram definidos os seguintes parâmetros genéticos:

- Tamanho da População (TP): tamanho da população existente em cada geração do AG, para buscar a solução; foram utilizadas três configurações, 20, 50 e 100 indivíduos.
- Taxa de Cruzamento (TC): taxa que indica o total de cruzamentos a ser realizado em cada geração do AG.
- NG: Número de gerações, ou seja, o total de iterações que o AG irá executar até gerar a população final e, consequentemente, a resposta para o problema.
- Taxa de Mutação (TM): taxa que indica o total de mutação a ser realizada em cada geração.
- Intervalo de Geração (IG): taxa que indica o total da população atual que será mantida a cada nova geração de descendentes.

4. TESTES E RESULTADOS

Os testes foram realizados de forma a experimentar a eficiência do AG ao trabalhar com diferentes configurações. Inicialmente, o objetivo do AG era organizar 20 pessoas em 4 grupos dadas suas configurações. Em seguida, aumentou-se o número de pessoas para 100 com o propósito de analisar seu desempenho, considerando um grupo maior.

Para este exemplo, utilizou-se uma situação de divisão de equipes genérica, com a necessidade de dividir seus componentes em equipes para a realização de um projeto. Os parâmetros de entrada utilizadas neste contexto foram gerados aleatoriamente, durante a execução do AG. Cada ponto de habilidade é avaliado perante uma nota, que pode variar de 0 a 10. A Tabela 1 apresenta uma amostragem dos resultados coletados durante os testes.

Tabela 1 | Amostragem dos resultados obtidos durante os testes.

ID	TP	NG	TC	TM	IG	Nº de Pessoas	Nº de Grupos	Resultado (1ª Geração)	Resultado (Última Geração)	Taxa de Melhora (%)
AG01	20	100	0,6	0,05	0,1	20	4	39	3	92,31%
AG02	20	100	0,7	0,15	0,3	20	4	43	3	93,02%
AG03	20	200	0,9	0,15	0	20	4	35	3	91,43%
AG04	50	200	0,8	0,15	0,2	20	4	33	3	90,91%
AG05	50	200	0,8	0,15	0,3	20	4	43	3	93,02%
AG06	100	50	0,7	0,15	0,1	100	4	96	4	95,83%
AG07	100	50	0,7	0,1	0,2	100	4	72	4	94,44%
AG08	100	50	0,6	0,1	0,2	100	4	92	6	93,48%
AG09	100	50	0,7	0,1	0,1	100	4	91	7	92,31%
AG10	100	50	0,6	0,2	0,1	100	4	81	7	91,36%

Como se pode observar mediante aos dados apresentados, o AG desenvolvido cumpre de forma satisfatória a necessidade de se trabalhar com diferentes configurações. Neste sentido, destacam os resultados da última geração, sempre próximos de 0, conforme apresentado na descrição da função de fitness. É interessante que, ao analisarmos a amostragem, a taxa de melhora em média é de 92,81%, proporcionando em todos os casos, uma configuração de grupos homogênea com relação ao nível médio de habilidades dentro de cada equipe.

5. CONCLUSÃO

Com base nos resultados apresentados na subseção anterior, o AG apresentado neste artigo colabora com a resolução do problema proposto, sendo capaz de cumprir a sua tarefa de formar as equipes o mais homogeneamente possível. Por ter sido desenvolvido de forma genérica, o mesmo pode ser aplicado em cenários de formação de equipes distintos, servindo assim a diversos propósitos, sendo apenas necessário ajustar as variáveis de entrada com a situação desejada.

No presente momento o AG não está habilitado para classificar equipes com tamanhos diferentes, a fim de se necessário, algumas equipes possam ter membros a mais que outras para tentar garantir que todos os grupos estejam o mais próximo possível da avaliação ideal.

REFERÊNCIAS

BEJARANO, V. C.; PILATTI, L. A.; LIMA, I. A. Equipes de Alta Performance. **Revista Tecnologia e Humanismo**. Curitiba. Vol. 19, n. 29 (2005), p. 23-34.

BLANCHARD, K. **The one minute manager builds high performance teams.** Bostom: Quill, 2000.

DRUCKER, P. **Administrando em tempos de grandes mudanças**. São Paulo: Pioneira/Thomsom Learning, 2001, p. 59-62.

HOLLAND, J. H. **Adaptation in natural and artificial Systems**. Ann Arbor, MI: University of Michigan Press, 1975.

HOLLAND, J. H. **Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence**. Cambridge, MA, USA: MIT Press, 1992.

HOLLAND, J. H.; GOLDBERG, D. E. **Genetic algorithms in search, optimization, and machine learning**. Massachusetts: Addison-Wesley, 1989.

KATZENBACH, J. R.; SMITH, D. K. **Equipes de alta performance: conceitos, princípios e técnicas para potencializar o desempenho das equipes**. Rio de Janeiro: Campus, 2001.

LARSON, C. E.; LARSON, C.; LAFASTO, F. M. J. **Teamwork: what must go right/what can go wrong**. Sage, 1989.

SILVA, Y. K. N. **A utilização de Algoritmos Genéticos para a formação de equipes heterogêneas em turmas universitárias**. Caicó, 2017.

SILVEIRA, S. R. Formação de grupos colaborativos em cursos a distância via web: um estudo de caso utilizando técnicas de Inteligência Artificial**. Revista brasileira de informática na educação**. Florianópolis. Vol. 14, n. 2 (maio-ago. 2006), p. 29-40.

SIVANANDAM, S. N.; DEEPA, S. N. Genetic algorithms. **In: Introduction to genetic algorithms**. Springer, Berlin, Heidelberg, 2008. p. 15-50.

STRNAD, D.; GUID, N. A fuzzy-genetic decision support system for project team formation. **Applied Soft Computing**, v. 10, n. 4, p. 1178-1187, 2010.

WALT, S. V. D.; COLBERT, S. C.; VAROQUAUX, G. The NumPy array: a structure for efficient numerical computation. **Computing in Science & Engineering**, v. 13, n. 2, p. 22-30, 2011.

PYTHON,ORG. Welcome to Python.org. Disponível em: https://www.python.org/.

WI, H. *et al*. A team formation model based on knowledge and collaboration. **Expert Systems with Applications**, v. 36, n. 5, p. 9121-9134, 2009.