

PLANEJAMENTO DE TRILHAS NO ENSINO DE ENGENHARIA: UMA PROPOSTA ATRAVÉS DE ALGORITMOS GENÉTICOS

José Tarcísio Franco de Camargo – jtfc@bol.com.br
Eliana Anunciato Franco de Camargo – eaftcamargo@yahoo.com
Centro Regional Universitário de Espírito Santo do Pinhal, UNIPINHAL
Av. Hélio Vergueiro Leite, s/n – Jardim Universitário
CEP 13990-000 – Espírito Santo do Pinhal – SP

Gilmar Barreto – gbarreto@dsif.fee.unicamp.br
Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação, UNICAMP, FEEC
Av. Albert Einstein, 400, Cidade Universitária Zeferino Vaz, Barão Geraldo
CEP 13083-852 – Campinas – SP

Jorge Cândido – jocandido@utfpr.edu.br
Universidade Tecnológica Federal do Paraná
Campus de Campo Mourão – UTFPR-CM
Departamento de Gestão e Economia – DAGEE
Vila Rosalina Maria dos Santos, 1.23
CEP – 87.301-899 – Campo Mourão – PR

Estéfano Vizconde Veraszto – estefanovv@cca.ufscar.br
Universidade Federal de São Carlos
Depto. de Ciências da Natureza, Matemática e Educação, UFSCar, CCA
Rodovia Anhanguera, Km 174
CEP 13604-900 – Araras - SP

Resumo: O planejamento de trilhas é um tipo clássico de problema que surge em diferentes campos da engenharia, como programação de rotas, prevenção de colisões, robótica e desenvolvimento de jogos. Por sua vez, por se tratar de um problema de otimização, a busca por caminhos ótimos pode se beneficiar do uso de algoritmos heurísticos, como os disponíveis na computação evolucionária. Assim, apresentar aos alunos aplicações para métodos não convencionais de otimização, tal como o uso de algoritmos genéticos, constitui uma oportunidade ímpar para o desenvolvimento de soluções inovadoras neste ramo da engenharia. Desta forma, este trabalho apresenta um procedimento para avaliar o melhor caminho entre dois pontos, em um ambiente restrito por obstáculos, através do uso de um algoritmo genético. A implementação desenvolvida começa buscando o percurso ideal a partir de trilhas geradas aleatoriamente, que evoluirão para a melhor solução possível. Os resultados demonstram a capacidade do algoritmo de aprender e produzir caminhos adequados sem conhecimento prévio do ambiente. Pode-se concluir que o algoritmo é simples, produz caminhos confiáveis e é rápido o suficiente para lidar com problemas em tempo real, permitindo que o mesmo seja aplicado em questões práticas.

Palavras-chave: Otimização. Computação Evolucionária. Robótica. Ensino de Engenharia.

1 INTRODUÇÃO

O planejamento de trilhas encontra aplicações nos mais diversos campos de estudo. Em cursos de engenharia são bastante comuns em robótica onde, por exemplo, procura-se determinar a melhor trajetória para o deslocamento de um robô (CHANDRA, 2015; COOK, 2011).

Neste sentido, este problema pode ser encarado através da combinação de duas perspectivas: determinar a menor distância entre dois pontos simultaneamente à prevenção de colisões com objetos presentes no ambiente. Os fundamentos da geometria euclidiana nos mostram que a menor distância entre dois pontos é definida pelo segmento de reta que os une. Contudo, em um ambiente restrito por obstáculos, esta simplicidade deixa de ser válida. Em nosso estudo, a abordagem proposta envolve o uso de algoritmos genéticos (AGs) para a determinação do melhor caminho entre dois ou mais pontos. A escolha desta estratégia deve-se ao fato de que a mesma possui uma implementação computacional relativamente simples, podendo inclusive vir a ser paralelizada em um cluster de computadores. O algoritmo aqui proposto também apresenta a vantagem de se adaptar a ambientes dinâmicos, onde outros objetos também podem se mover pelo espaço. Outra vantagem consiste no fato de que este tipo de algoritmo não exige o conhecimento prévio do ambiente, permitindo que o objeto em movimento descubra/aprenda o ambiente à medida em que se desloca.

Além disso, sob o ponto de vista educativo, em um curso de engenharia, o uso de AGs pode ser abordado como alternativa aos métodos convencionais de otimização. Em particular, além de problemas na área de robótica, esta estratégia se mostra bastante interessante para a abordagem de problemas estocásticos em geral (TU & LU, 2004).

As seções seguintes descrevem detalhadamente o procedimento desenvolvido para a aplicação de AGs no planejamento de trilhas.

2 DEFINIÇÕES TOPOLÓGICAS PARA O PROBLEMA

Previamente à introdução do algoritmo é importante apresentar as definições básicas que fundamentam o problema tratado. Neste sentido, a seguir são apresentadas as definições necessárias para o uso de um AG na determinação do melhor caminho entre dois pontos.

Mapas e trilhas

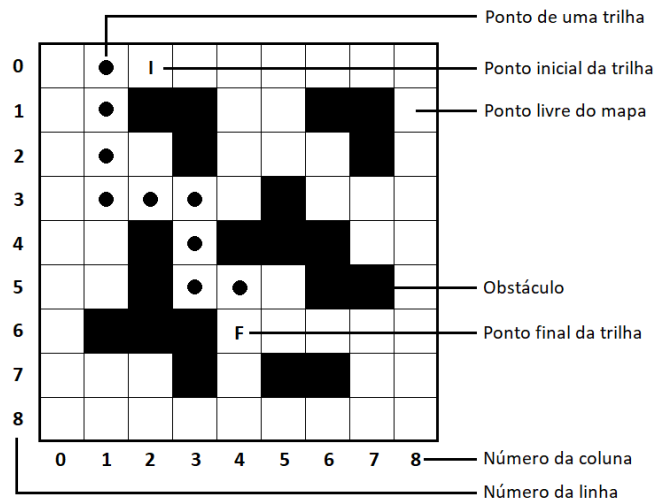
Um mapa consiste em um ambiente através do qual um objeto pode se deslocar. Em um mapa pode haver obstáculos que impedem a passagem do objeto por uma determinada região. No mapa também são indicados os pontos de início e fim de uma trajetória, além da trilha propriamente percorrida. A Figura 1 ilustra o conceito de mapa apresentado.

De maneira a organizar a utilização de um mapa, o mesmo pode ser interpretado como uma matriz de células com n linhas e m colunas, sendo que cada célula desta matriz pode conter um ponto livre, um obstáculo, o ponto inicial, o ponto final ou um ponto de passagem de uma trilha.

Por sua vez, uma trilha é composta por um conjunto de células de um mapa. Esta deve conter, obrigatoriamente, seu ponto de partida, seu ponto de chegada e os demais pontos percorridos pelo objeto. Numericamente, uma trilha consiste em um vetor que contém as coordenadas das células que a compõe. Para o exemplo da Figura 1, podemos expressar a trilha de acordo com a Equação (1).

$$Tr = [(0,2); (0,1); (1,1); (2,1); (3,1); (3,2); (3,3); (4,3); (5,3); (5,4); (6,4)] \quad (1)$$

Figura 1 – Estrutura de um mapa e seus componentes.



Fonte: Elaborado pelos autores.

Restrições para o deslocamento de um objeto

Como restrição ao movimento de um objeto, ao se deslocar em um mapa, este poderá apenas se mover nas direções “Norte”, “Sul”, “Leste” e “Oeste”.

3 FUNDAMENTOS DOS ALGORITMOS GENÉTICOS

Uma forma de determinar um possível caminho entre dois pontos consiste no uso de um algoritmo genético. Estes algoritmos possuem grande vocação para a solução de problemas de otimização, como é o caso do modelo aqui tratado. Dessa forma, um algoritmo genético pode ser construído de maneira a fornecer uma estimativa inicial para a solução do problema, ou seja, um caminho qualquer entre dois pontos específicos e, com o apoio de uma função objetivo e determinados procedimentos, refinar a solução inicial até ser encontrada a solução mais apropriada para o problema. O uso de algoritmos genéticos em problemas de otimização foi inicialmente proposto em 1975 (HOLLAND, 1992), tendo sido popularizado através de Goldberg (1989) e Haupt (2004). De maneira sucinta, pode-se afirmar que um algoritmo genético propõe uma analogia à Teoria da Evolução das Espécies de Charles Darwin (DARWIN, 2009), a qual, por sua vez, teve início com a integração de conceitos entre seleção natural e genética proposta por Gregor Mendel (MILLER, 2009). Assim, em um ambiente computacional, procura-se determinar a solução ótima para o problema a partir de estimativas iniciais quaisquer. Para tanto, o processo de otimização requer um “espaço de busca”, formado por um conjunto de “indivíduos” que compõem uma “população” de possíveis soluções; uma “função objetivo”, a qual direciona a busca por melhores soluções; e um conjunto de procedimentos, tais como “cruzamento” e “mutação”, que são utilizados no processo de otimização das soluções (BING, 2016; GUPTA 2016; KRAMER, 2017).

Assim, o uso de algoritmos genéticos implica em um processo estocástico, onde possíveis soluções são agrupadas formando uma população, sendo todos os indivíduos desta avaliados simultaneamente, com notas mais altas atribuídas aos mais “aptos”, ou seja, às melhores soluções. Por sua vez, a evolução da população em direção à solução otimizada passa por procedimentos onde os indivíduos combinam-se entre si, em processos de cruzamento, ou sofrem mutações, de

forma análoga ao que ocorre em populações biológicas. Tal evolução fomenta a origem de novas gerações, que devem representar melhores soluções para o problema abordado.

De forma genérica, este processo é descrito no algoritmo abaixo.

1. Se possível, deve ser gerada de maneira aleatória uma população de indivíduos que representam estimativas iniciais para a solução do problema tratado. Assim, a partir desta população inicial, que constitui um espaço de busca em direção à melhor solução, enquanto um determinado critério evolutivo não for atingido, devem ser repetidos os passos a seguir.
2. Avalia-se cada um dos indivíduos da população, atribuindo aos mesmos notas que representam suas respectivas “aptidões” para a solução do problema tratado. Tais notas são construídas a partir da função objetivo, que representa o referencial para a melhor solução. Quanto melhor a nota de um indivíduo, mais próximo ele está da solução ótima.
3. Selecionam-se os melhores indivíduos da população, para que estes possam se combinar aos pares, determinados por sorteio, em processos de cruzamento. Através destes cruzamentos, os indivíduos trocam parte de suas estruturas, dando origem a uma nova geração para a população.
4. Submete-se cada um dos indivíduos da nova geração a uma eventual “mutação”. Neste processo, parte de um determinado indivíduo pode sofrer alguma modificação, mediante a ocorrência de um evento de baixa probabilidade.
5. Aplica-se o “elitismo” na nova geração. Isto implica em trazer para a nova geração o melhor (ou os melhores) indivíduo(s) da geração atual, mantendo assim preservada a melhor solução obtida até o momento.
6. Executa-se novamente os passos de 2 a 5, até que sejam satisfeitos os critérios evolutivos esperados, ou seja, a população ou um de seus indivíduos atinja os limites da solução ótima.

A aptidão de um indivíduo x_i de uma população pode ser representada através de uma função $f(x_i)$, a qual indica quão próximo este indivíduo se encontra da solução ótima para o problema estudado. Assim, em uma população composta por N indivíduos, cada um destes terá a sua própria aptidão definida através de $f(x)$.

A análise e comparação das aptidões dos indivíduos de uma população irá estabelecer a probabilidade $p(x_i)$ que um indivíduo i terá de gerar descendentes, através do processo de cruzamento. Para o caso onde esta probabilidade é diretamente proporcional ao valor numérico de $f(x)$, então a mesma pode ser calculada através da Equação (2).

$$p(x_i) = \frac{f(x_i)}{\sum_{k=1}^N f(x_k)} \quad (2)$$

Caso a probabilidade aumente à medida em que o valor numérico da função objetivo tenda a zero, como é o caso do modelo discutido neste texto, então a probabilidade de seleção de um indivíduo deverá ser calculada conforme apresentado na Equação (3).

$$p(x_i) = \frac{1 - \left(\frac{f(x_i)}{\sum_{k=1}^N f(x_k)} \right)}{(N-1)} \quad (3)$$



4 ELEMENTOS PARA IMPLEMENTAÇÃO DO ALGORITMO GENÉTICO

O detalhamento do algoritmo apresentado na seção anterior requer, inicialmente, a construção da seguinte estrutura que representa um indivíduo de uma população de soluções.

Estrutura indivíduo

```
{  
Vetor de pontos trilha;  
Real aptidão;  
Real probabilidade de seleção;  
};
```

Nesta estrutura, “indivíduo” é um dos elementos da população que representa uma estimativa para a solução do problema tratado. Esta solução encontra-se representada pelo elemento “trilha” da estrutura, a qual contém um possível caminho entre os pontos de partida e chegada. Por sua vez, “aptidão” indica a qualidade desta solução, especificamente, a distância entre os pontos inicial e final da trilha. Assim, quanto menor o valor numérico de “aptidão”, ou seja, menor a distância entre os pontos, mais apta é a trilha. Finalmente, “probabilidade” indica a relevância da solução em relação aos demais indivíduos da população. Dessa forma, quanto maior o valor numérico de “probabilidade de seleção”, maior a chance deste indivíduo ser selecionado para reprodução.

4.1 Geração aleatória de uma trilha

Conforme mencionado previamente, a construção inicial de uma trilha pode ser realizada de forma aleatória. Assim, através de um algoritmo de “busca cega”, é construída uma proposta inicial para a trilha entre os pontos inicial e final. A sequência de passos apresentada abaixo demonstra uma proposta para o algoritmo de busca cega. Para interpretá-lo, considere a estrutura previamente apresentada na Figura 1.

1. Considere, inicialmente, que não é permitido se deslocar para fora dos limites do mapa ou para onde se encontra um “obstáculo” no mapa. Também não é permitido que um mesmo ponto seja percorrido mais de uma vez na trilha.
2. Faça o ponto inicial da trilha ser o ponto atual.
3. Se a partir do ponto atual não for possível se deslocar em nenhuma das quatro direções (Norte, Sul, Leste ou Oeste), volte ao Passo 2 deste algoritmo.
4. A partir do ponto atual, selecione aleatoriamente uma direção (Norte, Sul, Leste ou Oeste).
5. Se o ponto na direção selecionada estiver fora dos limites do mapa OU for parte de um obstáculo OU já fizer parte da trilha, volte ao Passo 4 deste algoritmo.
6. Faça o ponto que se encontra na direção selecionada ser o ponto atual.
7. Se o ponto atual for o ponto final da trilha, encerre o algoritmo; caso contrário, volte ao Passo 3.

É importante esclarecer que o algoritmo de busca cega apresentado acima pressupõe que existe um caminho que possa ser percorrido desde o ponto inicial até o ponto final da trilha. As restrições inicialmente impostas no algoritmo impedem que a trilha atravesse um obstáculo presente no mapa ou saia fora de seus limites, bem como evita a construção de trilhas contendo laços, ou seja, a passagem repetitiva por pontos do mapa.

Embora o algoritmo anterior seja capaz de apresentar um caminho aleatório entre dois pontos sem a presença de laços, sua natureza não impede que trechos irrelevantes sejam percorridos. Considere, por exemplo, quatro pontos consecutivos: k , $k+1$, $k+2$ e $k+3$. Se $k+3$ for geograficamente vizinho (ao norte, sul, leste ou oeste) do ponto k , então os pontos $k+1$ e $k+2$ devem ser removidos da trilha e $k+3$ deve se tornar o ponto subsequente a k .

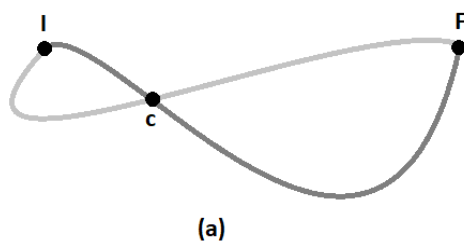
4.2 Cruzamento e mutação

Cruzamento e mutação são operações utilizadas em algoritmos genéticos com o intuito de buscar uma solução mais eficiente que a atual. Em uma operação de cruzamento, duas trilhas que possuem pelo menos um ponto em comum (exceto os pontos inicial e final) permutam trechos entre si, buscando com isso obter trilhas menores. Em uma operação de mutação, uma trilha altera parte de seu trajeto, preferencialmente reduzindo seu tamanho.

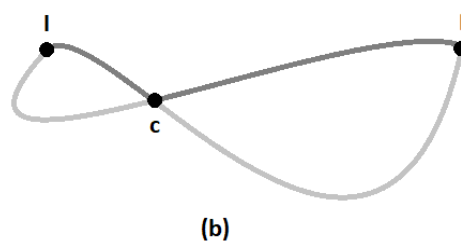
A seleção de duas trilhas de uma população para cruzamento é determinada através de um sorteio, baseado na aptidão de cada indivíduo. Dessa forma, quanto mais apto o indivíduo, ou seja, quanto mais curta a trilha, maior a probabilidade do indivíduo ser sorteado para cruzamento.

A Figura 2 ilustra a operação de cruzamento entre duas trilhas.

Figura 2 – (a) Trilhas antes do cruzamento.



(b) Trilhas após o cruzamento.



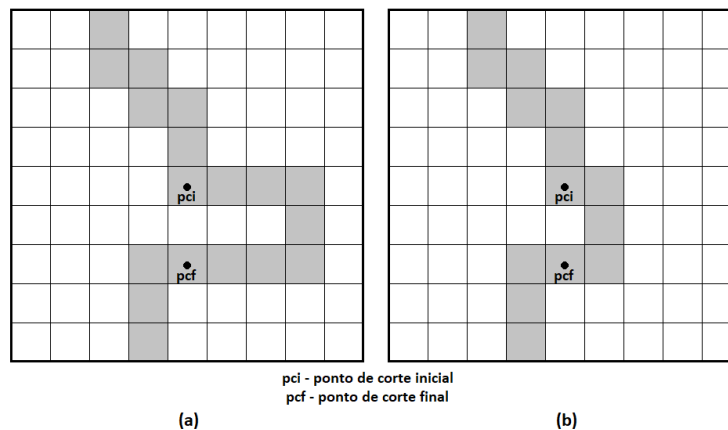
Fonte: Elaborado pelos autores.

Conforme apresentado nas figuras 2(a) e 2(b), no procedimento de cruzamento as trilhas permutam seus segmentos que vão desde o ponto de interseção (c) até o ponto final das trilhas. Como pode haver mais de um ponto de interseção, neste caso, o ponto de corte deverá ser escolhido aleatoriamente. Em uma população, as trilhas que passarão por cruzamento serão selecionadas por sorteio, tendo como base a probabilidade de seleção de cada trilha. O algoritmo abaixo apresenta uma síntese da operação de cruzamento.

1. Considere duas trilhas que possuem pelo menos um ponto em comum, exceto os pontos inicial e final.
2. Caso as trilhas possuam dois ou mais pontos de interseção, selecione aleatoriamente um destes pontos.
3. Para cada uma das duas trilhas, a partir do ponto selecionado, permuta os trechos destas trilhas que vão desde o ponto de interseção até o ponto final destas.

Por sua vez, a Figura 3 ilustra o procedimento de mutação em uma trilha.

Figura 3 – (a) Trilha antes da mutação. (b) Trilha após a mutação.



Fonte: Elaborado pelos autores.

Observe a presença de dois pontos de corte na trilha. Entre estes dois pontos será realizada a mutação. A escolha do primeiro ponto de corte deve ser aleatória, sendo o segundo ponto selecionado em função de uma determinada distância em relação ao primeiro. Empiricamente, recomenda-se que uma mutação seja um evento de baixa probabilidade (até 5%) e que não afete mais do que 10% da trilha, de forma a evitar a degeneração da população. Também recomenda-se que o procedimento de mutação não dê origem a uma trilha maior que a inicial. O algoritmo a seguir apresenta uma síntese da operação de mutação, que possui grande semelhança com o algoritmo de busca cega apresentado anteriormente. Considere ser ainda válida a restrição de que a trilha não pode passar por um determinado ponto mais de uma vez.

1. Determine, por sorteio, se a trilha deve sofrer mutação. Se a trilha não deve passar por mutação, encerre este procedimento.
2. Determine, por sorteio, o ponto de corte inicial da mutação.
3. Determine o ponto de corte final da mutação.
4. Faça o ponto de corte inicial da mutação ser o ponto atual.
5. Se a partir do ponto atual não for possível se deslocar em nenhuma das quatro direções (Norte, Sul, Leste ou Oeste), volte ao Passo 4.
6. A partir do ponto atual, selecione aleatoriamente uma direção para o movimento (Norte, Sul, Leste ou Oeste).
7. Se o ponto na direção selecionada estiver fora dos limites do mapa ou for parte de um obstáculo, volte ao Passo 6.
8. Faça o ponto que se encontra na direção selecionada ser o ponto atual.
9. Se o ponto atual for o ponto de corte final, encerre o algoritmo; caso contrário, volte ao Passo 5.

4.3 Elitismo

Devido à sua estrutura heurística e à presença de eventos aleatórios, os procedimentos de cruzamento e mutação, por si só, não garantem a convergência do algoritmo em direção a uma solução ótima da geração atual para a seguinte. Dessa forma, torna-se interessante preservar, de uma geração para a outra, os melhores indivíduos da geração atual. Este procedimento é conhecido como “elitismo”. Assim, devem ser separados os melhores indivíduos da geração atual e, após a

realização das operações de cruzamento e mutação, os indivíduos mais aptos da geração atual devem substituir os indivíduos menos aptos que foram obtidos através de cruzamento e mutação.

Finalmente, a sequência de passos abaixo apresenta em detalhes o algoritmo genético desenvolvido neste trabalho.

1. Obtenha, aleatoriamente, “ N ” indivíduos para a 1ª geração da população. Estes indivíduos devem ser obtidos a partir do algoritmo de busca cega.
2. Determine a aptidão de cada indivíduo, através do cálculo da distância percorrida desde o ponto inicial até o ponto final da trilha.
3. Calcule a probabilidade de seleção de cada indivíduo da população através da Equação (3).
4. Encontre o indivíduo com maior aptidão nesta geração.
5. Enquanto o indivíduo mais apto não satisfizer o critério de parada:
 - a. Para iniciar uma nova geração, repita “ $N/2$ ” vezes:
 - i. Selecione, por sorteio, com base em sua probabilidade de seleção, dois indivíduos da geração atual.
 - ii. Realize o cruzamento dos dois indivíduos selecionados.
 - iii. Armazene os dois indivíduos gerados por cruzamento na nova geração.
 - iv. Fim.
 - b. Submeta todos os indivíduos da nova geração a um eventual processo de mutação.
 - c. Selecione os indivíduos menos aptos da nova geração e substitua-os pelos indivíduos mais aptos da geração atual.
 - d. Calcule a aptidão de cada indivíduo da nova geração.
 - e. Calcule a probabilidade de seleção de cada indivíduo da nova geração.
 - f. Selecione o indivíduo mais apto da nova geração.
 - g. Faça a nova geração ser a geração atual.
 - h. Fim.
6. Fim.

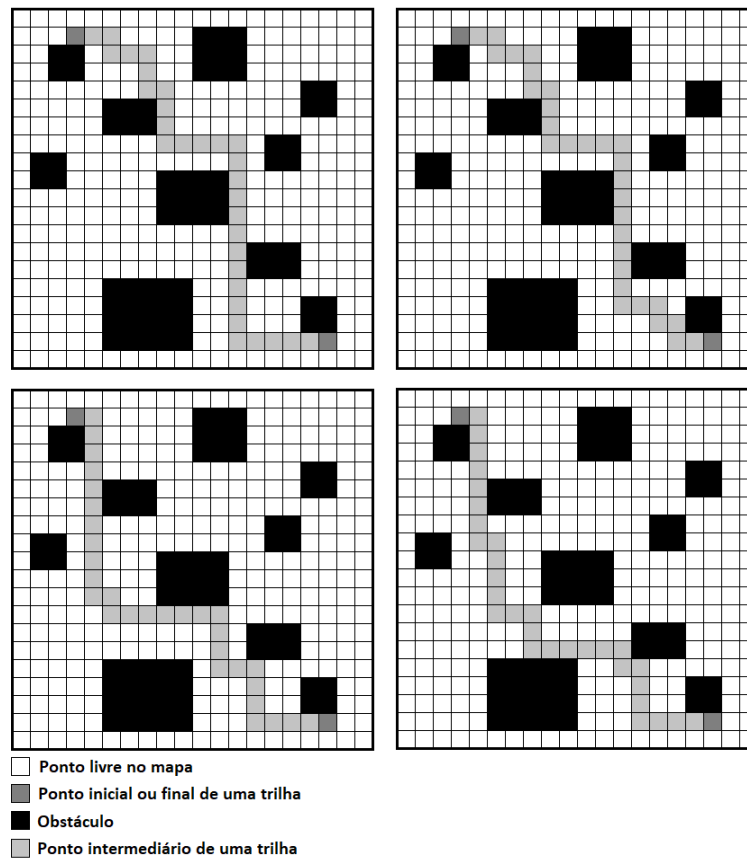
5 RESULTADOS OBTIDOS

O modelo aqui discutido foi simulado em um mapa organizado na forma de uma matriz de 20 linhas por 20 colunas, conforme indicado na Figura 4. As trilhas apresentadas nesta figura foram geradas a partir de simulações com uma população de 20 indivíduos, sendo a taxa de mutação igual a 2%. Uma mutação, caso ocorresse, não poderia afetar mais do que 10% de uma trilha.

Para a inicialização do algoritmo foram admitidas trilhas geradas aleatoriamente, com um comprimento inicial de, no máximo, 80 unidades, sendo que cada unidade equivale a uma célula percorrida no mapa. Como critério de parada para a simulação foram admitidas trilhas com, no máximo, 32 unidades de comprimento. A escolha de 32 unidades de comprimento como critério de parada deve-se ao fato de que, para os pontos inicial e final definidos neste mapa, a menor distância possível entre eles equivale a um caminho com exatamente 32 unidades.

Embora não seja possível garantir que um AG sempre irá alcançar a solução ótima, os resultados obtidos através destas simulações demonstram que esta convergência pode ser alcançada. Além disso, os resultados também mostram a existência de mais de uma solução ótima para o problema, demonstrando a diversidade de soluções que um AG é capaz de fornecer.

Figura 4 – Trilhas geradas através da simulação do modelo apresentado.



Fonte: Elaborado pelos autores.

6 CONSIDERAÇÕES FINAIS

Os resultados deste estudo demonstram que o uso de algoritmos genéticos para a geração de trilhas otimizadas é viável. Uma vantagem considerável para o uso de AGs neste tipo problema consiste em sua implementação computacional relativamente simples. O algoritmo apresentado também é paralelizável, podendo ser fragmentado em um cluster de computadores, diminuindo o tempo de cálculo para a obtenção de soluções mais precisas em ambientes com restrições mais severas de movimento. Em relação à velocidade de convergência do algoritmo, esta depende fortemente da quantidade de indivíduos presentes em uma população. De fato, através das simulações executadas, observa-se empiricamente que utilizando poucos indivíduos a convergência para a solução do problema não ocorre ou ocorre apenas depois de um número relativamente grande de gerações. Por sua vez, populações maiores tendem a promover a convergência do algoritmo, sendo conveniente ressaltar que em grandes populações a convergência é mais lenta. Outra vantagem significativa no uso de AGs em problemas de otimização consiste na sua alta flexibilidade, o que os torna facilmente adaptáveis para outros tipos de problemas de otimização.

REFERÊNCIAS

BING, X.; MENGJIE, Z.; BROWNE, W. N. A Survey on Evolutionary Computation Approaches to Feature Selection. **IEEE Transactions on Evolutionary Computation**, v. 20, n. 4, p. 606-626, 2016.

- CHANDRA, Mohanta J. **Introduction to Mobile Robots Navigation**. USA: Lambert Academic Publishing, 2015.
- COOK, Gerald. **Mobile Robots: Navigation, Control and Remote Sensing**. USA: Wiley-IEEE, 2011.
- DARWIN, Charles. **A origem das Espécies e a Seleção Natural**. São Paulo: Madras, 2004.
- GOLDBERG, David E. **Genetic Algorithms in Search, Optimization and Machine Learning**. Boston: Addison-Wesley Longman Publishing Co., 1989.
- GUPTA, A.; YEW-SOON, O.; LIANG, F. Multifactorial Evolution: Toward Evolutionary Multitasking. **IEEE Transactions on Evolutionary Computation**, v. 20, n. 3, p. 343-357, 2016.
- HAUPT, Randy L.; HAUPT, Sue E. **Practical Genetic Algorithms**. 2nd. Edition, New Jersey: John Wiley & Sons, 2004.
- HOLLAND, John H. **Adaptation in natural and artificial systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence**. Cambridge, MA: A Bradford Book (MIT Press), 1992.
- KRAMER, Oliver. **Genetic Algorithm Essentials**. New York: Springer, 2017.
- TU, Z.; LU, Y. A Robust Stochastic Genetic Algorithm (StGA) for Global Numerical Optimization. **IEEE Transactions on Evolutionary Computation**, v. 8, n. 5, p. 456-470, 2004.
- MILLER, Frederic P.; VANDOME, Agnes. F.; McBREWSTER, John. **Gregor Mendel**. London: Alphascript Publishing, 2009.

PATH PLANNING IN ENGINEERING EDUCATION: A PROPOSAL THROUGH GENETIC ALGORITHMS

Abstract: *Path planning is a classic type of problem that arises in different engineering fields, such as route planning, collision avoidance, robotics, and game development. In turn, because it is an optimization problem, the search for optimal tracks can benefit from the use of heuristic algorithms, such as those available in evolutionary computation. Thus, introduce students to applications for unconventional optimization methods, such as the use of genetic algorithms, is a unique opportunity for the development of innovative solutions in this field of engineering. In this way, this work presents a procedure to evaluate the best path between two points, in an environment restricted by obstacles, through the use of a genetic algorithm. The developed implementation starts by searching for the ideal route from randomly generated tracks that will evolve to the best possible solution. The results demonstrate the ability of the algorithm to learn and produce adequate paths without prior knowledge of the environment. It can be concluded that the algorithm is simple, produces reliable paths, and is fast enough to handle problems in real time, allowing it to be applied in practical issues.*

Key-words: Optimization. Evolutionary Computing. Robotics. Engineering Teaching.