

Algoritmo genético para planejamento de caminhos baseados em curvas de Bézier

Thiago Meireles Grabe

Resumo—O presente trabalho aborda o planejamento de caminhos para robôs holonômicos através de curvas de Bézier e algoritmo genético. Neste problema, a tarefa de planejar uma trajetória possível considerando uma posição inicial e final e os possíveis obstáculos é apresentada com uma abordagem estocástica do algoritmo evolucionário e a parametrização das curvas de Bézier. Operadores genéticos elitistas são utilizados para a evolução da população (possíveis caminhos). Simulações de cenários são realizadas para a validação da aplicabilidade da solução proposta e, ao final, alguns resultados são apresentados e discutidos bem como as vantagens e limitações desta abordagem.

I. INTRODUÇÃO

Em robótica o planejamento de trajetórias é um campo amplamente estudado devido à importância e alcance hoje em dia, haja vista os robôs em ambientes industriais e mesmo domésticos. Um exemplo é o setor de manufatura que pode utilizar robôs que aumentam a produção de larga escala além da qualidade do processo. Para robótica móvel, estes ativos são extremamente úteis em ambientes hostis como zonas de contaminação. Dada a importância dos robôs atividades diversas, o planejamento de caminhos é essencial para atingir um determinado alvo e desviar de obstáculos, garantindo assim o propósito para o qual foi desenvolvido.

Planejar um caminho pode não ser uma tarefa simples dada a complexidade do ambiente em que se deseja realizar uma atividade e, ao mesmo tempo, envolve deliberar muitas vezes considerando ambientes estáticos. Como premissa fundamental, conhecer o ambiente é fundamental para um planejamento ótimo. O planejamento de caminhos (PC) pode ser classificado em planejamento reativo ou local e deliberativo ou global. O PC reativo envolve uma alta resolução próximo à obstáculos o que auxilia os desvios de pequenos objetos na trajetória ao alvo. Por ser reativo e local, esta técnica demanda uma tomada de decisão em tempo real. Já o PC deliberativo usualmente gera trajetórias de alto nível evitando obstáculos ao longo do caminho, mas é sensível à resolução do mapa que representa o ambiente. Algumas características importantes devem ser consideradas ao realizar o PC: Complexidade, completude e eficácia são fundamentais para que o planejamento seja bem sucedido.

A proposta do presente trabalho é atuar de forma deliberativa sobre o planejamento de trajetórias utilizando um algoritmo genético (AG), baseado em curvas de Bézier, para gerar caminhos ótimos do ponto de partida até a posição final, evitando obstáculos em diversos cenários de configurações. Por fim, um conjunto de configurações para o robô é apresentada como solução para o PC observando a curvatura do caminho e as restrições físicas do robô.

Com esta abordagem, o presente trabalho propõe uma solução viável para deliberar em ambientes complexos de diversos obstáculos considerando o conhecimento prévio do ambiente através de um mapa. Portanto, uma solução para que o robô possa navegar por ambientes hostis e completar tarefas em um mapa.

II. TRABALHOS RELACIONADOS

O PC é um problema que vem sido abordado por uma diversidade de trabalhos da área de robótica móvel. Em [4] é feita uma análise de abordagens clássicas e recentes abordagens evolucionárias, como algoritmos genéticos. Cada forma de entender o problema de PC é vista como ótima em determinados cenários, mas devem levar em consideração o tempo computacional, a eficiência para evitar obstáculos e a distância de deslocamento entre o alvo e o ponto de partida.

A representação do ambiente é parte fundamental para se planejar caminhos. Alguns algoritmos são baseados em mapas geométricos ou topológicos em que um ambiente contínuo e transformado em um mapa discreto. Pode-se citar alguns algoritmos como *Road map* que identifica rotas dentro do *free space*, decomposição de células que faz uma separação/classificação do ambiente discretizado em células vazias ou ocupadas. Por fim, o algoritmo de campos potenciais impõe uma formulação matemática de atração ao alvo e repulsão à obstáculos no espaço [10]. Contudo, há algoritmos que consideram decisões locais em que o foco é chegar ao alvo desviando de obstáculos a partir de dados de sensores por exemplo. Neste contexto pode-se citar os *Bug Algorithms* que são algoritmos simples e possuem uma característica de circular o obstáculo, mas podem ser ineficientes em determinados ambientes [2], [10].

AGs são apresentados como soluções viáveis para PC em [6], [5], [3]. Tais algoritmos foram desenvolvidos na década de 60 e são baseados em uma abstração biológica da evolução de cromossomos. Através de operadores genéticos como *Crossover* (cruzamento) e *Mutation* (mutação) uma população inicial é evoluída até uma função de custo chamada *Fitness* ser atendida e uma resposta ótima ser apresentada [7].

Alguns trabalhos como em [12] um AG é apresentado em um ambiente discretizado em que a distância euclidiana e a ocupação das células são importantes para a composição da função de *fitness*. [6] e [1] também apresentam um AG baseado em curvas de Bézier. Estes autores tocam em um ponto importante que é a curvatura realizada pela trajetória. Essa restrição devem ser observadas e consideradas em funções de *fitness*, pois ao lidar com certos tipos de robôs,

como *Unmanned Aerial Vehicles* (UAVs) é fundamental observar as restrições físicas do veículo utilizado.

O presente trabalho não apenas visa um AG que apresenta o melhor caminho baseado em uma função de *fitness* para minimizar o caminho ou evitar obstáculos, mas apresenta uma abordagem em que ambos são considerados além de se observar a curvatura da curva de Bézier gerada para que o caminho seja factível com as restrições físicas do robô. A representação do indivíduo neste trabalho difere dos trabalhos mencionados, pois a representação dos genes em [6] se dá pelos pontos de controle das curvas de Bezier enquanto o trabalho desenvolve os cromossomos pelas configurações assumidas pelo robô. Em suma, este trabalho considera um mapa conhecido como premissa para uma abordagem deliberativa do PC, um robô holonômico e um AG com operadores elitistas para planejar uma trajetória válida.

III. FORMULAÇÃO DO PROBLEMA

Uma trajetória realizado por um robô pode ser definida como uma sequencia de configurações que ele assume durante um intervalo de tempo. Ir de uma posição *Start* para uma posição *Goal* requer um planejamento prévio. Cada sequencia de configurações assumida ao longo do tempo pelo robô será um elemento ou indivíduo do AG. Assim, ao se conhecer o mapa com os obstáculos os principais problemas a serem resolvidos por este trabalho são:

- 1) Determinar, de forma estocástica, as configurações válidas para que o robô se desloque de uma posição *Start* para uma posição *Goal*, evitando os obstáculos através de um AG;
- 2) Atribuir a cada sequencia de configurações um valor de *fitness* em função dos parâmetros:
 - a) Curvatura do trajeto;
 - b) Distância euclidiana;
 - c) Obstáculos.
- 3) Realizar um *tuning* dos parâmetros do AG para que o algoritmo seja completo para os testes propostos.

IV. METODOLOGIA

A técnica apresentada tem como premissas o conhecimento do ambiente através de um mapa e as restrições cinemáticas para a trajetória dada uma curvatura mínima do robô. As configurações do ponto de partida (P_i) e chegada (P_c) do robô são conhecidas. Um caminho é estabelecido pela curva de Bézier através de uma curva paramétrica $\mathbf{b}(t)$ em um plano em que o parâmetro t varia em \mathbb{R} . Sendo assim, cada configuração da trajetória é definida da seguinte forma:

$$P_n(x_n, y_n) = \mathbf{b}(t_n) \quad (1)$$

cujos valores de t_n são os n valores para a curva de Bézier $\mathbf{b}(t)$.

Cada valor de (x_n, y_n) representa uma configuração a ser assumida pelo robô na trajetória. As posições (P_i) e (P_c) são fixas em cada cenário de teste.

A. Curvas de Bézier e Curvatura da trajetória

Curvas de Bezier são curvas paramétricas polinomiais no plano ou no espaço. Elas podem descrever curvas e superfícies de forma simples, o que as torna extremamente eficientes dados os pontos de controle ao estabelecer uma série de segmentos de retas ligando os pontos.

O i -ésimo coeficiente da curva é obtido através do binômio de Newton e é representado como um polinômio da forma descrita na equação 2:

$$P_i(t) = \binom{n}{i} (1-t)^{n-i} \cdot t^i \quad (2)$$

Com isso, pode-se definir a expressão para a curva de Bézier $\mathbf{b}(t)$:

$$\mathbf{b}(t) = \sum_{i=0}^n P_i(t) \cdot B_i = \sum_{i=0}^n \binom{n}{i} (1-t)^{n-i} \cdot B_i \quad (3)$$

Na equação 3 o termo B_i representa os pontos de controle da curva de Bézier. Neste trabalho são cinco pontos de controle, sendo os pontos que o ponto inicial e final são fixados (*start* e *goal*) e os demais escolhidos aleatoriamente.

Pontos de Controle	Representação
C_0	Posição Inicial
C_1	Aleatório
C_2	Aleatório
C_3	Aleatório
C_4	Posição Final

Tabela I. Pontos de controle da Curva de Bezier

No planejamento de trajetórias curvas suaves favorecem determinadas tarefas delicadas, e neste contexto as curvas de Bézier são importantes. Todavia, a derivada de segunda ordem é fundamental para essa suavidade do caminho. De acordo com [11], essa derivada garante a curvatura da curva de Bezier, sendo assim um ponto de avaliação do caminho. A função que define uma curvatura em um espaço n dimensional é descrita pela equação 4:

$$\kappa = \frac{\|\vec{r}' \times \vec{r}''\|}{\|\vec{r}'\|^3} \quad (4)$$

B. Algoritmo Genético

Essa parte do trabalho aborda a modelagem do algoritmo genético, seus operadores e a sua representação do indivíduo no contexto de PC.

1) *Cromossomo*: Cada indivíduo do AG é um candidato possível para a solução final do problema do caminho a seguir executado pelo robô. Após os pontos de controle serem definidos, uma curva de Bezier é estabelecida e cada ponto no plano representa uma configuração do robô em que as posições (x, y) são os genes deste cromossomo.

$$\begin{aligned} P_0 &= (x_0, y_0) \\ P_1 &= (x_1, y_1) \\ P_{f-1} &= (x_{f-1}, y_{f-1}) \\ P_f &= (x_f, y_f) \end{aligned} \quad (5)$$

Em que (x_0, y_0) e (x_f, y_f) são as posições, inicial e final, respectivamente do caminho. Os demais genes do cromossomo são escolhidos durante a geração da curva de Bezier na inicialização da população. O indivíduo é representado pelas configurações do caminho em que cada gene é uma configuração do robô.

(x_0, y_0)	(x_1, y_1)	(x_2, y_2)	...	(x_{f-1}, y_{f-1})	(x_f, y_f)
--------------	--------------	--------------	-----	----------------------	--------------

Figura 1. Representação do Indivíduo

2) *Função de fitness*: A função de *fitness* em um AG é essencial para o apelo evolucionário do algoritmo assim como os operadores genéticos. Os seguintes aspectos serão abordados na função:

- (a) Função de energia considerando restrições do robô;
- (b) Distância euclidiana entre os pontos da trajetória;
- (c) Obstáculos no mapa.

Em [8] é proposta uma função que relaciona as restrições físicas do robô em um espaço tridimensional:

$$\omega = \sqrt{\left(\frac{\kappa(t)}{\kappa_{max}}\right)^2 + \left(\frac{\tau(t)}{\tau_{max}}\right)^2 + \left(\frac{\theta(t)}{\theta_{max}}\right)^2} \quad (6)$$

Na equação (6) as restrições são representadas pela curvatura máxima κ_{max} , torsão máxima τ_{max} , além do ângulo máximo de mergulho ou decolagem ω_{max} . Dado que este trabalho aborda problemas em um plano bidimensional, apenas a parcela da curvatura máxima será considerada através da equação (7).

$$\omega_p = \left(\frac{\kappa(t)}{\kappa_{max}}\right) \quad (7)$$

A distância euclidiana entre os pontos gerados pela curva de Bézier será considerada na função (8) de *fitness*:

$$\Lambda = \sum_{i=1}^i \lambda_i = \sqrt{(x_{i-1} - x_i)^2 + (y_{i-1} - y_i)^2} \quad (8)$$

Em que i são as posições do caminho e essa distância é calculada ponto a ponto na curva.

Por fim, dado um obstáculo os pontos que representam as configurações do robô não podem fazer parte de um obstáculo. Em linhas gerais, caso o trajeto possua algum ponto $P = (x, y)$ que é um obstáculo, a função penaliza este indivíduo do AG.

$$\phi = \begin{cases} \alpha, & \text{é obstáculo,} \\ 0, & \text{caminho válido} \end{cases} \quad (9)$$

Na equação (9), α é uma constante definida como um parâmetro do AG para se obter diversidade na população, haja vista que soluções ruins, a priori, podem gerar boas soluções após os operadores genéticos.

Por fim, a função *fitness* será:

$$\Upsilon = \sqrt{\omega_p^2 + \Lambda^2 + \phi^2} \quad (10)$$

3) *Operadores Genéticos*: Após iniciado o AG, uma população inicial é gerada e possíveis soluções para o problema são avaliadas. Neste trabalho foi abordado o operador de cruzamento e o de mutação genética para aumentar a diversidade da população e melhorar a procura pelas melhores soluções dentro do espaço de busca, evitando assim uma convergência prematura.

Contudo, esses operadores foram modelados de forma elitista. Esta decisão se baseia que o processo de tomada de decisão por um robô para deliberar sobre um caminho não pode ser totalmente independente da variável tempo. Dito isto, um caminho ótimo deve ser encontrado, mesmo sendo um ótimo local.

A figura (2) demonstra a ação dos operadores genéticos no valor da função *fitness* do AG. O cruzamento genético realiza a troca do material genético de cada indivíduo, o que localmente ao decorrer da evolução do AG, tende a encontrar um mínimo na região do espaço de busca. Este processo pode encontrar um mínimo local ou global. Neste tocante, a variabilidade genética dos indivíduos gerados é fundamental para uma ampla busca no espaço de busca das soluções. Já a mutação, por ter um caráter mais agressivo nos indivíduos, tende a alterar a região do espaço de busca. Este operador pode ser fundamental para que o AG saia de regiões de mínimos locais e possa fazer uma busca por um mínimo global. Vale ainda ressaltar que caso a mutação ocorra de forma exagerada, a convergência do AG pode ser comprometida [9].

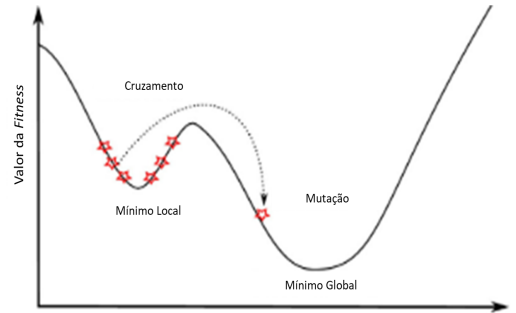


Figura 2. Operadores genéticos na busca ótima. Adaptado de [1]

O cruzamento tem a função de diversificar a população em um subespaço para a convergência neste subespaço amostral. Neste processo exemplificado na figura 3, seleciona-se os cromossomos através das 50 melhores soluções através da *fitness* e dentre esse subconjunto, dois cromossomos são combinados trocando partes de seus genes. Os novos indivíduos gerados, ou cromossomos filhos, são então avaliados e, caso atendam o critério de seleção, são incorporados à população de soluções candidatas. Caso contrário, são descartadas. O cruzamento direciona a população a uma convergência no subespaço a uma solução ótima, haja vista que ao decorrer do tempo as soluções tendem a este ponto ótimo.

Já a mutação é um operador que atua de forma probabilística na população. Dada uma probabilidade de ocorrência, um

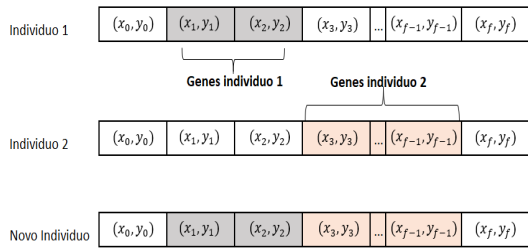


Figura 3. Cruzamento

indivíduo é selecionado da população. Um gene é escolhido deste indivíduo e uma nova configuração é inserida no lugar do gene escolhido. A figura 4 O processo de mutação pode auxiliar o AG uma busca por todo espaço de soluções, mas caso seja realizado de forma exagerada pode dificultar a convergência da solução.

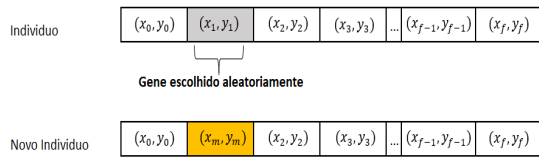


Figura 4. Mutação

4) *Pseudocódigo*: O método proposto neste trabalho consiste em gerar uma população de forma aleatória, dado os pontos de início e fim do caminho a ser desenvolvido pelo robô. Após esta inicialização, os cromossomos irão sofrer operações genéticas e, através de uma métrica definida, a melhor solução dessa geração será apresentada como caminho possível para o robô seguir da posição inicial à sua posição alvo.

O pseudocódigo (1) para este trabalho demonstra algumas decisões para o PC. Dado que é um algoritmo elitista, somente soluções viáveis são retornadas como *Melhor solução*. Todavia, o critério de parada é atingido somente quando esta *Melhor solução* atinge um patamar mínimo estabelecido como *fitness mínima*.

Não obstante, os operadores genéticos são acionados enquanto o critério de parada não for atingido. Esta decisão acarreta benefícios e alguns problemas de mínimos locais e convergência prematura.

V. EXPERIMENTOS

Neste trabalho os experimentos foram realizados em linguagem *Python* e as simulações em *ROS*.

Foram utilizados dois mapas de tamanho fixo (16x16) com obstáculos. A figura 5 mostra esses mapas e como os obstáculos estão dispostos. O primeiro mapa representa um ambiente com múltiplos obstáculos enquanto o segundo mapa um ambiente com um obstáculo de tamanho considerável em relação ao tamanho total do mapa.

As curvas de Bezier geradas necessitam de alguns parâmetros para que o caminho planejado, ou curva gerada, tenha a sua curvatura suavizada. O número de pontos de

Entrada: P_0 , P_f , n obstáculos, Gerações, *fitness mínima*, Probabilidade de mutação

Resultado: Conjunto de configurações do robô entre *Start: P_0* e *Goal: P_f*

```

1  $P_0 \leftarrow Start$ 
2  $P_f \leftarrow Goal$ 
3  $g \leftarrow Gerações$ 
4  $f \leftarrow fitness\ mínima$ 
5  $m \leftarrow Probabilidade\ de\ mutação$ 
6  $best \leftarrow \emptyset$ 
7 foreach  $i \in g$  do
8   Gera-se, aleatoriamente, uma população inicial com
    $n$  soluções candidatas  $\chi = [S_0, S_1, ..., S_n]$ 
9   foreach  $S \in \chi$  do
10     $Y(S); \#Avalia\ População$ 
11     $Graded \leftarrow sort(\chi); \#Ordenação\ crescente$ 
12    if  $Y(Graded[0]) \leq Y(best)$  then
13       $best \leftarrow Graded[0]$ 
14    end
15  end
16  while  $Y(best) \leq f$  do
17    Evoluir  $Graded$ 
18     $\gamma = rand(0,1)$ 
19    if  $\gamma \leq m$  then
20       $mutação()$ 
21    end
22    else
23       $cruzamento()$ 
24    end
25     $Y(Graded); \#Avalia\ Nova\ População$ 
26     $Graded \leftarrow sort(Graded); \#Ordenação\ crescente$ 
27    if  $Y(Graded[0]) \leq Y(best)$  then
28       $best \leftarrow Graded[0]$ 
29    end
30  end
31 end
32 return  $best$ 
33

```

Algoritmo 1: AG Elitista para Planejamento de Caminhos

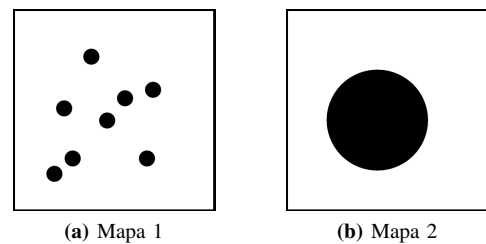


Figura 5. Mapas

controle está diretamente relacionado com as mudanças de trajetória haja vista que a curva é paramétrica. Outro ponto é o número de pontos em cada curva. Este último é igual ao número de genes de cada indivíduo do AG.

Os parâmetros do AG foram determinados de forma

empírica. Devido a esse fato, tais parâmetros foram distintos para cada mapa. A tabela II demonstra os parâmetros utilizados no experimento.

Parâmetro	Valores
Número de Indivíduos	150
Número de Gerações	350
Probabilidade de mutação	10%
Probabilidade de cruzamento	90%
Quantidade de genes	50
Parcela $\alpha(fitness)$	1000

Tabela II. Parâmetros Algoritmo Genético

As curvas de Bezier geradas necessitam de alguns parâmetros para que o caminho planejado, ou curva gerada, tenha a sua curvatura suavizada. O número de pontos de controle está diretamente relacionado com as mudanças de trajetória haja vista que a curva é paramétrica. Outro parâmetro importante do trabalho é o número de pontos em cada curva. Este último é igual ao número de genes de cada indivíduo do AG. A tabela III apresenta esses parâmetros:

Parâmetro	Quantidades
Quantidade de pontos de controle	5
Quantidade pontos na curva	50

Tabela III. Parâmetros da Curva de Bezier

Para todos os testes realizados a tabela IV mostra os pontos inicial (*start*) e final (*goal*). Neste caso, esses pontos são coincidentes com o primeiro e último ponto de controle das curvas de Bezier geradas.

Pontos	Configuração (X,Y)
Start	[0, 0]
Goal	[15, 15]

Tabela IV. Configuração Inicial e Final dos experimentos

A. Experimento Mapa 1

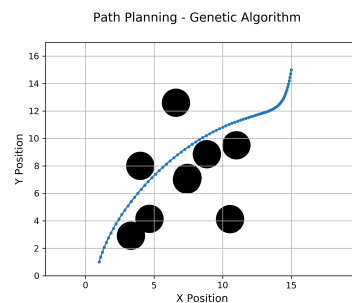
No primeiro teste realizado, o mapa foi desenvolvido com vários obstáculos localizados em pontos aleatórios do plano. Ao todo foram nove obstáculos e pode ser identificado na figura 6 a trajetória do melhor indivíduo em um dos testes realizado bem como a convergência do AG ao longo das gerações:

É possível notar que a convergência para o PC se dá entre as gerações 300 e 350. Além disso, nota-se a característica elitista dos operadores genéticos ao observarmos o rápido decaimento da *fitness* no início do algoritmo e, ao mesmo tempo, a demora em convergir para um valor próximo a zero.

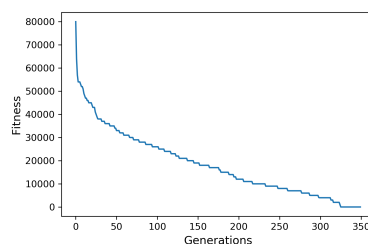
Apesar da deliberação ter certo custo em relação ao tempo, percebe-se que a trajetória foi criada evitando-se os obstáculos seguindo a curva de Bezier com a curvatura comportada.

B. Experimento Mapa 2

Já no primeiro teste realizado, o mapa foi desenvolvido com apenas um obstáculo mas proporcionalmente grande em relação ao mapa. A figura mostra 7 a trajetória do melhor indivíduo em um dos testes realizado bem como a convergência do AG ao longo das gerações:

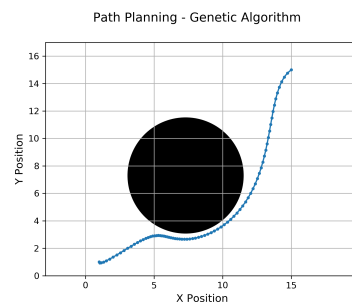


(a) Mapa 1

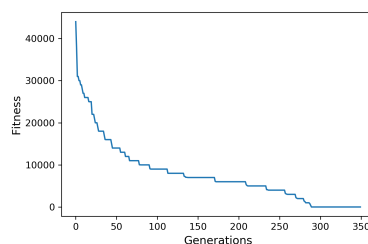


(b) Fitness ao longo das Gerações

Figura 6. Experimento 1



(a) Mapa 2



(b) Fitness ao longo das Gerações

Figura 7. Experimento 2

VI. CONCLUSÕES

Ao realizar os dois experimentos utilizando o mesmo algoritmo em mapas com configurações distintas, é possível observar que o AG é de fato capaz de gerar trajetórias possíveis. Tanto nos resultados do experimento V-A quanto no experimento V-B observa-se que trajetórias foram definidas evitando-se obstáculos de forma efetiva.

Um ponto importante dos experimentos é o tempo de convergência em função das gerações. Como os operadores

genéticos são elitistas era esperado um tempo maior de convergência em relação aos trabalhos abordados. Contudo, acredita-se que o elitismo no início do AG pode ser amenizado utilizando técnicas para uma melhor exploração do espaço de busca, seja através de um torneio menos elitista ou por meio do *Stochastic universal sampling*. Além disso, o presente trabalho modela os indivíduos como sendo as configurações do robô ao longo da trajetória. Esse método pode ser ineficiente nos operadores genéticos haja vista que somente configurações que são concorrentes em algum ponto da trajetória podem preencher os pré requisitos dos operadores de cruzamento. Isso pode explicar o número de gerações para a convergência da solução.

As premissas adotadas neste trabalho abordam o conhecimento do mapa, odometria conhecida e assume um ambiente estático. Essa premissa é forte haja vista que ambientes com essas características e um total conhecimento do mapa por vezes pode ser utópico. Com isso, alguns problemas não tratados neste trabalho precisam ser levantados como erros na odometria em função dos modelos cinemáticos utilizados, simplificações na modelagem dos obstáculos que podem gerar cenários de colisão. Contudo, vale ressaltar que o presente estudo é importante por tratar um paradigma deliberativo e pode validar técnicas importantes como a utilização de AG neste contexto e a importância dos operadores genéticos que permitem tanto uma busca abrangente no espaço de configurações quanto a convergência para soluções ótimas.

Como o mapa possui tamanho fixo, a exploração do AG está diretamente relacionada com o tamanho do mapa. Sendo assim, um mapa de maior tamanho em ambas as coordenadas pode não ter o contorno dos obstáculos bem sucedido pelo AG devido ao número de pontos de controle da curva de Bezier. Dito isso, aumentar o número destes pontos de controle aumenta a maleabilidade da trajetória. Além dos pontos de controle, o número de configurações para cada indivíduo pode não ser suficiente o que tornaria necessário estabelecer uma relação entre o tamanho do mapa e esses dois parâmetros descritos.

Em suma, o trabalho apresentou uma solução válida para o planejamento de trajetórias para ambientes que se tem conhecimento a priori. Abordar conhecimentos da disciplina de Robótica Móvel e aplicar em um arcabouço de algoritmo evolucionário foi bastante desafiador dada a não certeza de convergência para soluções de fato válidas.

VII. TRABALHOS FUTUROS

Para trabalhos futuros pode-se abordar robôs diferenciais ou mesmo *Unmanned Aerial Vehicles* (UAVs). Neste contexto o número de genes do indivíduo seria aumentado e a complexidade do AG maior. Contudo, pode-se criar uma solução viável para deliberar trajetórias para esses tipos de robôs.

A representação do indivíduo foi feita através das 50 configurações ao longo da trajetória. Uma possível abordagem é também representar o indivíduo pelos pontos de controle da curva de Bezier. Isso permitiria um operador de cruzamento mais efetivo ao longo das gerações.

Ainda sobre a representação do indivíduo, foi levantado uma situação em que os mapas deste trabalho apresentam tamanho fixo. Uma direção importante para futuros trabalhos e estabelecer uma possível relação entre o tamanho do mapa e quantidade de configurações necessárias para descrever bem a trajetória. Além disso, uma relação entre o número de pontos de controle e as características do mapa para que se tenha caminhos ótimos.

Por fim, a função de *fitness* [10] pode ser aprimorada para que características sejam melhores exploradas. Por exemplo, a distância é normalizada pela distância entre os pontos inicial e final. Pode-se propor outro método para se ponderar esta parcela.

REFERÊNCIAS

- [1] Mohamed Elhoseny, Alaa Tharwat, and Aboul Ella Hassanien. Bezier Curve Based Path Planning in a Dynamic Field using Modified Genetic Algorithm, 2018.
- [2] Umair Hashmi, Farhana Afshan, and Maliha Rafiq. Performance analysis of different optimal path planning algorithms on a client server based mobile surveillance UGV, 2013.
- [3] B.-Y. Kang, J. Lee, and D.-W. Kim. Fast genetic algorithm for robot path planning, 2013.
- [4] Weria Khaksar, Khairul Salleh, Mohamed Saharia, Firas B Ismail, and Universiti Tenaga Nasional. A Review on Mobile robots Motion Path Planning in unknown environments, 2015.
- [5] Yi Liu, Ming Cong, Hang Dong, Dong Liu, and Yu Du. Time-Optimal Motion Planning for Robot Manipulators Based on Elitist Genetic Algorithm, 2017.
- [6] Douglas G. Macharet, Armando Alves Neto, and Mario Fernando Montenegro Campos. Feasible UAV Path Planning Using Genetic Algorithms and Bezier Curves. pages 223–232, 2010.
- [7] Melanie Mitchell. Genetic Algorithms : An Overview 1, 1995.
- [8] Armando A. Neto and Mario F.M. Campos. On the generation of feasible paths for aerial robots with limited climb angle, 2009.
- [9] Riccardo Poli, W.B. Langdon, and N.F. McPhee. A Field Guide to Genetic Programming, 2008.
- [10] Roland Siegwart and Illah R Nourbakhsh. Introduction to Autonomous Mobile Robots, 2004.
- [11] Baoye Song, Zidong Wang, and Li Sheng. A new genetic algorithm approach to smooth path planning for mobile robots, 2016.
- [12] Mohamed Amine Yakoubi and Mohamed Tayeb Laskri. The path planning of cleaner robot for coverage region using Genetic Algorithms, 2016.