Sistema Dinâmico de Planejamento de Trajetória para Um Dirigível

Fábio Silveira Vidal
Instituto Militar de Engenharia - IME
Praça General Tibúrcio, 80 – Praia Vermelha – Rio de Janeiro – RJ - Brasil
fsvidal@de9.ime.eb.br

Júlio César Silva Neves
Instituto Militar de Engenharia - IME
Praça General Tibúrcio, 80 – Praia Vermelha – Rio de Janeiro – RJ - Brasil
sneves@de9.ime.eb.br

Paulo Fernando Ferreira Rosa
Instituto Militar de Engenharia - IME
Praça General Tibúrcio, 80 – Praia Vermelha – Rio de Janeiro – RJ - Brasil
rpaulo@de9.ime.eb.br

Resumo

Este trabalho descreve um sistema dinâmico de planejamento de trajetória para um dirigível, em vôos de altitude praticamente constante. Neste sistema, o objetivo é encontrar um caminho livre de obstáculos entre os pontos de partida e de chegada, de forma a minimizar o tempo gasto, pelo dirigível, para realizar tal percurso. Para tanto, foram utilizados parâmetros relacionados com a cinemática do dirigível para o cálculo da função objetivo, que retorna uma estimativa do tempo necessário para percorrer um determinado percurso. Para calcular o planejamento da trajetória, modela-se o ambiente usando o método de decomposição em células; e foi desenvolvido um algoritmo baseado na meta-heurística Colônia de Formigas, do qual, o resultado é melhorado através de otimizações. O sistema descrito neste artigo, funciona de forma dinâmica, ou seja, ele adapta a trajetória a ser percorrida conforme a ocorrência de mudanças na configuração do espaço de trabalho do dirigível enquanto um determinado percurso é percorrido pelo mesmo.
Palavras-Chaves: Colônia de formigas, planejamento de trajetória

Abstract

This paper describes a trajectory planning dynamic system for a bimp, in flights at an practically constant altitude. In this system, the goal is to find a obstacles free path between the goal and starting points, of form to minimize the time expense, for the blimp, to carry such passage. For this, blimp kinematics related parameters was used in objective function calculation, this function returns a time necessary estimate to cover one determined passage. To calculate the trajectory planning, the environment is modelled using the Cell Decomposition Method; and was developed an algorithm based on the Ant Colony meta-heuristic where the results are improved through optimizations. The described system in this paper possess dynamic functions form, i.e., it adapt the trajectory according the changes in the blimp workspace configuration while the passage is covered.
Keywords: Ant colony, trajectory planning

1 INTRODUÇÃO

O problema de planejamento de trajetória pode ser definido como a tarefa de conduzir um sistema móvel de uma posição corrente a uma posição de destino [9]. E é a atividade básica da robótica móvel; podendo, ainda, ser dividido em duas abordagens: (a) Global e (b) Local. No planejamento de trajetória global traça-se um caminho da posição inicial até a posição final. Tal

trajetória é calculada a partir de um mapa do ambiente armazenado previamente, supondo que este seja estático. É usado, principalmente, em navegação de longa distância onde os sensores do robô não conseguem captar informações sobre todo o percurso a partir da posição inicial. Caso apareçam novos obstáculos, no caminho definido inicialmente, este é modificado por meio de um planejamento local para que não aconteça colisão do atuador com eventuais obstáculos. Entre os métodos de planejamento de trajetória podemos citar: (a) Roadmaps; (b) Diagramas de Voronoi; (c) Campo Potencial; e (d) Campo de Força Virtual. Os métodos citados, podem necessitar de um tempo de processamento considerável ou apresentar problemas de mínimos locais. Como o problema resolvido neste trabalho possui restrição de tempo, optou-se por um processo heurístico que apesar de não retornar, sempre, uma solução ótima, fornece um bom resultado em um tempo de processamento aceitável.

Antes da aplicação de um método para determinação de um caminho, deve-se realizar o mapeamento do ambiente, para o cumprimento desta tarefa, neste trabalho, utiliza-se o método de Decomposição Celular. Este consiste em dividir o ambiente em células ou regiões. Esta divisão pode ser de forma exata ou aproximada. Feito o mapeamento, o próximo passo é calcular a combinação ótima (ou próxima da ótima) de transições entre as células para sair de um ponto inicial e chegar a um ponto final (objetivo). Neste trabalho esta combinação foi calculada por meio do método heurístico Colônias de Formigas.

A meta-heurística Colônia de Formigas foi originalmente proposta por Marco Dorigo[3] e é inspirada no comportamento de enxames, isto é, no comportamento coletivo de colônias de insetos, como as colônias de formigas, ou em sociedades de outros animais, funcionando com a idéia da comunicação indireta explorada pelas sociedades de insetos, que formam algoritmos distribuídos de multi-agentes[2].

Este trabalho está organizado da seguinte maneira: a próxima seção trata de alguns trabalhos relacionados ao assunto abordado neste trabalho; na seção 3 é descrito o processo de mapeamento do ambiente, feito pelo método da Decomposição Celular; a seção 4 descreve o cálculo do custo de um determinado caminho; na seção 5, aborda-se como é determinado um caminho; A seção 6 descreve os resultados obtidos ao final do trabalho, seguida pelas conclusões, trabalhos futuros e das referências bibliográficas.

2 TRABALHOS RELACIONADOS

Um trabalho que é bem relacionado com este é o de Thrun [11] que fez uma representação do ambiente dividindo-o em células e cada célula recebe uma probabilidade de ocupação, formando um mapa de probabilidades e, a partir dele, é feito o cálculo da trajetória. Em 2001 [10] estendeu o modelo de [11] de um ambiente 2D para um ambiente 3D em um trabalho para o planejamento de trajetória de um helicóptero. Para a escolha do melhor caminho da trajetória global, [10] usou o algoritmo Djikstra. Em [4] é encontrada uma abordagem sobre este algoritmo. Neste trabalho, consegue-se encontrar o caminho ótimo de um ponto ao outro, porém, ele utiliza um algoritmo exato que possui um custo computacional elevado, podendo não ser interessante para aplicações dinâmicas e com restrições de tempo.

Em [8] é descrito um método de planejamento de trajetória para dois robôs em um mesmo espaço de trabalho; O método em questão produz trajetórias livres de obstáculos e que não entram em estado de colisão. O trabalho [5] descreve um método de navegação híbrida em um ambiente conhecido e dinâmico para uma miniatura de um robô escalador; Este sistema de navegação calcula uma trajetória ótima a partir dados já conhecidos e adapta ao longo de sua execução com o uso de lógica fuzzy e redes neurais; Entretanto, este sistema de navegação apresenta situações de mínimos locais. O trabalho [6] faz uma abordagem sobre planejamento de trajetória para robôs móveis usando computação DNA. Um outro trabalho que utiliza a meta-heurística Colônia de Formigas é [1]. Diferentemente deste trabalho, [1] mapeia o ambiente por meio de um grafo e deposita o

feromônio em suas arestas.

3 MAPEAMENTO DO AMBIENTE

Apesar dos dirigíveis se deslocarem no espaço tridimensional, foi considerado um ambiente bidimensional por que pretende-se aplicar o método de planejamento de trajetória, explicado neste trabalho, em situações em que o dirigível atue em altitude praticamente constante. O ambiente de trabalho é divido usando o método de decomposição em células, uma abordagem sobre este método é encontrada em [7]. Desta forma, divide-se o ambiente de trabalho em células retangulares e de tamanhos iguais. As células são distribuídas em *n* linhas e *n* colunas formando um mapa geográfico de forma matricial. Este mapa é representado, matematicamente, por uma matriz binária. As células onde se encontram alguma parte de um obstáculo qualquer é representada pelo valor 1 e as demais representadas pelo valor 0. Assim, um caminho é uma seqüência de células livres, ou seja que possuam valor 0, e adjacentes. A seguir será exposto como é feito o cálculo do custo de um caminho.

4 CUSTO DE UM CAMINHO

O custo de um caminho é o valor correspondente à estimativa do tempo que o dirigível gastaria para sair do ponto de partida em estado de repouso e chegar ao ponto destino de forma a voltar ao repouso. Chamaremos esse tempo de "tempo de viagem". Para estimá-lo, são considerados os seguintes parâmetros relativos à cinemática do dirigível e ao mapeamento do ambiente: (a) tamanho das células; (b) velocidade de cruzeiro (velocidade máxima que pode ser alcançada pelo dirigível); (c) aceleração; (d) tempos de rotação (tempos gastos, em média, pelo dirigível para realizar as rotações de 45°, 90º, 135º e 180°).

O caminho gerado é formado por uma seqüência de seguimentos de retas, unidos por suas extremidades. Para calcular o custo de um caminho, primeiro determina-se o tempo que o dirigível levaria para ir de uma extremidade a outra de cada seguimento de reta que forma o caminho. Em seguida, é feito o somatório dos tempos gastos para fazer a rotação em cada transição de um segmento de reta para o próximo, no trajeto em questão. Para finalizar, são adicionados os tempos gastos para rotacionar o dirigível de sua pose inicial para a direção do primeiro seguimento de reta, chamada de rotação inicial; e para rotacionar o dirigível da direção do último seguimento de reta para a orientação de sua pose final. O custo total do percurso é a soma dos tempos de viagem de cada seguimento de reta mais o somatório do tempo total para fazer todas as rotações durante o percurso, incluindo as rotações inicial e final. Assim, podemos calcular o custo de um caminho através da seguinte equação:

$$Custo = [\sum_{i=1}^{ns} Ts(v, a, S_i) + \sum_{i=1}^{nr} G(R_i)] + G(ri) + G(rf) \quad (1)$$

onde *ns*: número de segmentos de reta que formam o caminho; *Ts*: função que retorna o tempo necessário para percorrer um segmento de reta, definida na equação 2. *v*: velocidade de cruzeiro. *a*: aceleração. *S*: vetor que contém os segmentos de reta que formam o caminho. *nr:* número de rotações entre os segmentos de reta que compõem o caminho. *G*: função que retorna o tempo gasto em determinada rotação. *R*: vetor que contém o número de graus referentes às rotações realizadas durante o percurso. *ri*: número de graus referente à rotação inicial. *rf*: número de graus referente a rotação final.

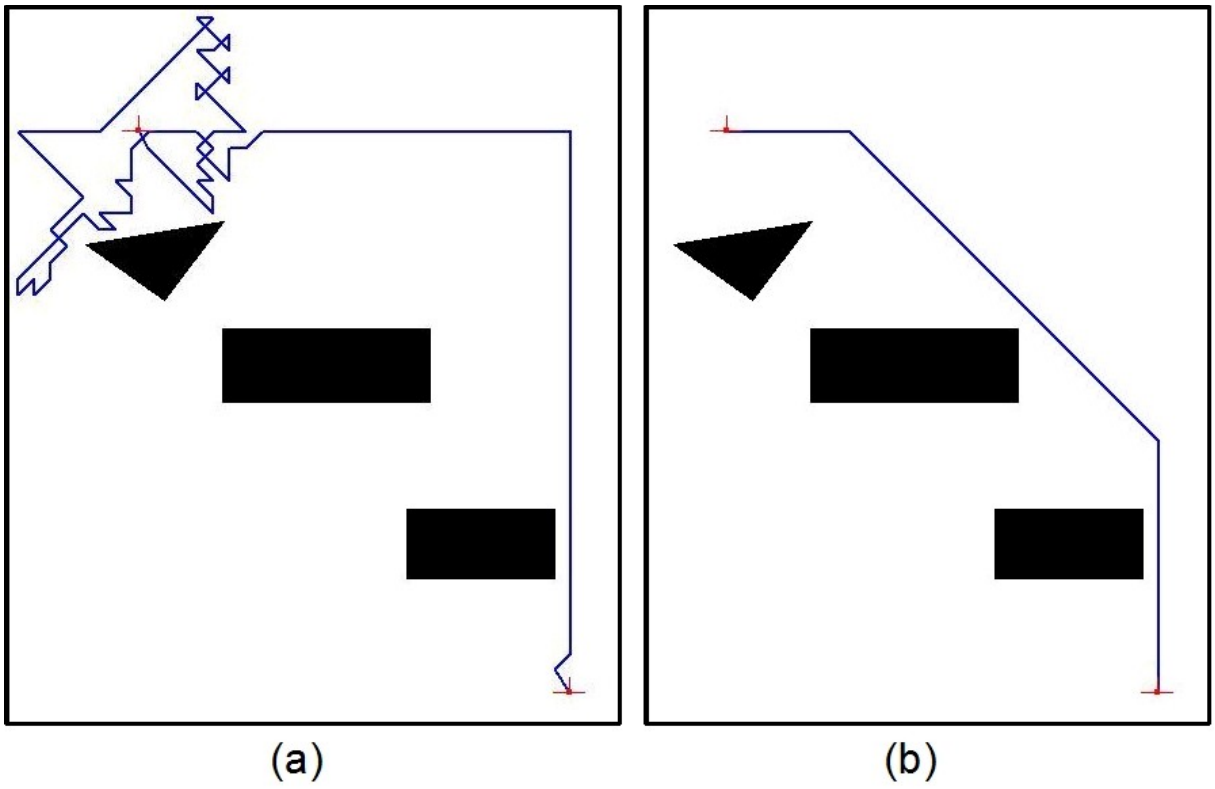

Figura 1: (a) Caminho gerado pela meta-heurística Colônia de Formigas. (b) Caminho de (a) após as otimizações na vizinhança e nas retas

$$Ts(v, a, c) = \begin{cases} 2\,x\,\dfrac{\sqrt{\dfrac{v^2}{a}}}{a} + c - \dfrac{v}{a}, \; se\; c > \dfrac{v^2}{2a} \\ 2\,x\,\dfrac{\sqrt{\dfrac{v^2}{a}}}{a}, \; se\; c \leqslant \dfrac{v^2}{2a} \end{cases} \qquad (2)$$

Cada caminho é obtido através de um método heurístico de planejamento de trajetória que é descrito a seguir.

5 DETERMINAÇÃO DE UM CAMINHO

O espaço de trabalho, onde atua o dirigível é dinâmico, ou seja, os obstáculos podem estar em movimento e é possível que surjam novos obstáculos durante o percurso, desta forma, o planejamento da trajetória deve ser feito de forma contínua. A partir deste ponto, surge uma restrição de tempo. Assim, optou-se pela utilização de um processo heurístico para determinar o caminho a ser seguido, que apesar de não retornar o caminho ótimo em todos os casos, forneça um resultado viável em tempo de processamento hábil. Portanto, usa-se a meta-heurística Colônia de Formigas [3], onde o depósito de feromônio está localizado em cada célula. O objetivo é fazer com que as formigas artificiais sigam na direção da célula mais próxima do objetivo.

Antes da primeira iteração, são feitas algumas trilhas iniciais de feromônio para que o ponto objetivo seja encontrado com maior rapidez. Tais trilhas correspondem às linhas, colunas e diagonais dos pontos de origem e de destino. Em seguida o processo da meta-heurística Colônia de Formigas é disparado. A cada iteração, de forma alternada, metade das formigas tentam ir do ponto

de saída ao ponto de chegada e a outra metade tenta fazer o caminho inverso. É estabelecido um número máximo de passos que uma formiga pode efetuar. As formigas que alcançam o número máximo de passos são desconsideradas. Uma outra restrição é que as formigas não podem passar por uma mesma célula mais de uma vez, isto é, o caminho não deve conter células repetidas.

Somente com o uso da meta-heurística colônia de formigas não foi possível conseguir bons resultados, observando a ilustração da figura 1(a) nota-se que os caminhos gerados não são diretos. Porém, observou-se que por meio de otimizações a solução pode ser melhorada. Então, a cada caminho gerado no processo, são feitas duas otimizações: (a) Otimização na vizinhança; (b) Otimização nas retas. A seguir serão descritas estas otimizações.

Inicialmente, notou-se que nos caminhos gerados uma mesma célula possuira mais de um vizinho pertencente ao caminho gerado. Assim, ao passar por uma célula qualquer, pode-se seguir para um de seus vizinhos, passar por outras células, e depois chegar a um outro vizinho da mesma célula. Assim, o caminho possui percursos desnecessários. Para resolver este problema foi desenvolvida a otimização na vizinhança, que consiste em verificar se cada célula possui mais de um vizinho pertencente ao caminho. Em caso afirmativo, escolhe-se o vizinho correspondente ao ponto mais adiante do caminho como próxima célula. Desta forma, os outros vizinhos e todas as outras células por onde o caminho iria passar de forma desnecessária são eliminados. Após a execução da otimização na vizinhança, nota-se que o caminho gerado ainda pode ser melhorado. O motivo é que ele tende a fazer um *zig-zag*. A otimização nas retas tenta eliminar os *zig-zags* deixados pela otimização na vizinhança. Esta operação é feita da seguinte forma: cria-se um novo caminho onde a primeira célula coincide com a primeira célula do caminho original, e a célula seguinte é a vizinha na direção em que se encontra a célula mais adiante pertencente ao caminho original, e que seja alcançável em linha reta (vertical, horizontal ou diagonal), ou seja, que esteja na mesma linha, coluna ou diagonal e com o percurso livre de obstáculo(s). Após o cálculo da segunda célula, determina-se as células seguintes com o mesmo processo até encontrar a última célula do caminho. A Figura 1(b) mostra um caminho gerado e otimizado pelos dois processos já citados. Note que ele possui uma trajetória bem mais comportada, ou seja, possui menos curvas. O pseudocódigo do algoritmo de determinação de um caminho é mostrado na figura 2.

6 RESULTADOS

Este trabalho descreveu um sistema de planejamento de trajetória global para dirigíveis em vôos em altitudes praticamente constantes. Foi implementado um simulador de planejamento de trajetória dinâmico em um espaço de trabalho bidimensional (figura 3), onde os obstáculos se movimentam e o caminho é adaptado conforme acontecem mudanças na configuração do espaço de trabalho. A figura 4 exemplifica uma seqüência de adaptações no caminho durante a execução da trajetória. No simulador, é possível ajustar parâmetros relativos à meta-heurística colônia de formigas (número de iterações, número de formigas por iteração, velocidade de evaporação do feromônio, número de passos por formiga, quantidade de feromônio depositada nas trilhas iniciais e a compensação do feromônio feita a cada iteração na trilha do melhor caminho encontrado) usada no planejamento de trajetória e à cinemática do dirigível (velocidade de cruzeiro, aceleração, tempo médio das rotações de 45º, 90º, 135º e 180º). A precisão do mapeamento do ambiente depende do número de linhas e colunas em que se divide o ambiente. Os obstáculos inseridos no ambiente de trabalho podem ser estáticos ou se moverem em um determinada direção, e podem possuir forma retângular, elíptoca ou a forma de um polígono qualquer. Também é possível inserir obstáculos durante a execução da trajetória fazendo com que o caminho seja adaptado de acordo com o surgimento de novos obstáculos, e com a movimentação dos obstáculos já inseridos.

```
algoritmo procurar_caminho(XOrigem, YOrigem, XDestino, Ydestino, NIteracoes, NpassosFormiga, NFormigasIteracao,
                           MatrizMapa, CustoMaximo)
        espalhar_feromonio(XOrigem, YOrigem, MatrizMapa, FTemp)
        espalhar_feromonio(XDestino, YDestino, MatrizMapa, FTemp)
        fazer_trilha(XOrigem, YOrigem, XDestino, YDestino, MatrizMapa, FTemp)
        sair ← 0
        n_celulas_caminho ← 0
        n_caminhos ← 0
        enquanto(i < NIteracoes e sair = 0)
                i++
                para j ← 0 até j < Nlinhas
                        para k ← 0 até k < NColunas
                                F[j][k] ← FTemp[j][k];
                        fim para
                fim para
                para j = 0 até j < Nformigas_Iteracao
                        se sair = 1
                                parar
                        fim se
                        se (resto(j, 2) <> 0)
                                XObjetivo ← XOrigem
                                YObjetivo ← YOrigem
                                x ← XDestino
                                y ← YDestino
                        fim se
                        senao
                                x ← XOrigem
                                y ← YOrigem
                                XObjetivo ← XDestino
                                YObjetivo ← YDestino
                        fim senao
                        n_celulas_caminho ← 0
                        enquanto (x <> x_objetivo ou y <> y_objetivo) e n_celulas_caminho < NPassosFormiga)
                                escolhe_proxima(x, y, MatrizMapa, F)
                                FTemp[y][x] ←  FTemp[y][x] + 0.5
                                caminho[j][0] ← x
                                caminho[j][1] ← y
                                n_celulas_caminho ← n_celulas_caminho + 1;
                        fim enquanto
                        se  x = XObjetivo e y = YObjetivo
                                n_caminhos ← n_caminhos + 1
                                otimizacao_retas(caminho, n_celulas_caminho)
                                otimizacao_vizinhanca(caminho, n_celulas_caminho)
                                se n_caminhos = 1
                                        melhor_caminho ← caminho
                                        n_celulas_melhor_caminho ← n_celulas_caminho
                                fim se
                                senao se custo(caminho, n_celulas_caminho) < custo(melhor_caminho,
                                                                        n_celulas_melhor_caminho)
                                        melhor_caminho ← caminho
                                        n_celulas_melhor_caminho ← n_celulas_caminho
                                fim senao se
                                para uk ← 0 até uk < n_celulas_caminho
                                        FTemp[caminho[uk][1]][caminho[uk][0]] ←
                                                                FTemp[caminho[uk][1]][caminho[uk][0]] + 1
                                fim para
                                se custo(caminho, n_celulas_caminho) <= CustoMaximo
                                        sair ← 1
                                fim se
                        fim se
                fim para
        fim enquanto
fim função
```

Figura 2: pseudocódigo do algoritmo de determinação de um caminho

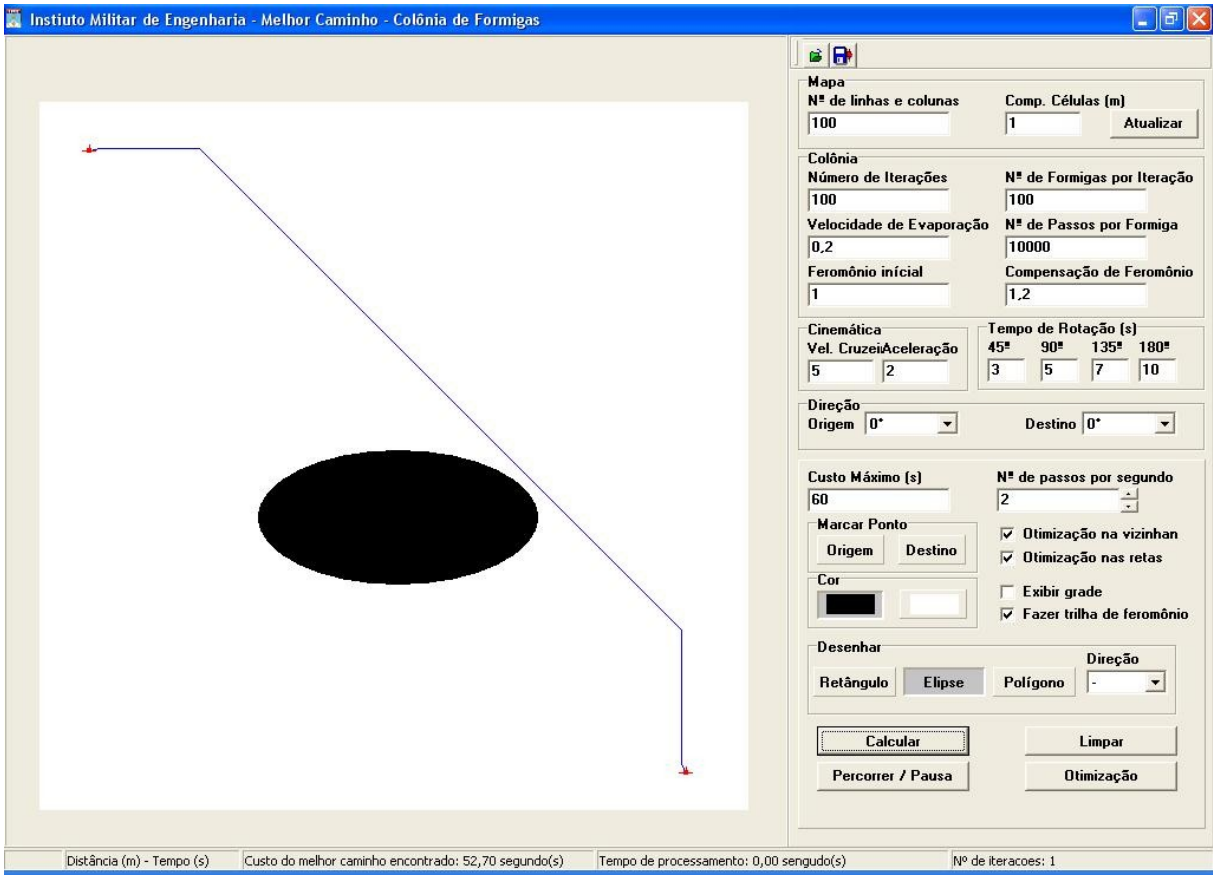

Figura 3: Simulador de planejamento de trajetória no plano bidimensional

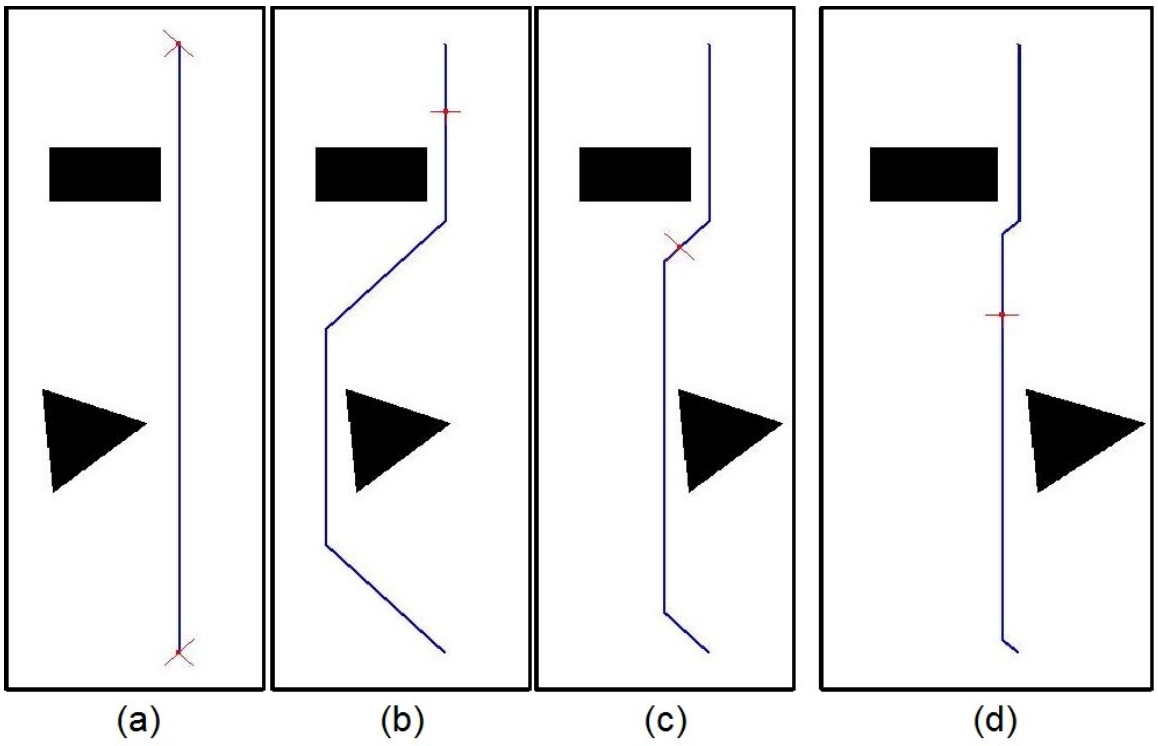

Figura 4: Seqüência de adaptações da trajetória

7 CONCLUSÕES

A meta-heurística Colônia de Formigas funciona de forma aleatória, sendo guiada apenas pela quantidade de feromônio depositada em cada célula, no caso deste trabalho. Por este motivo, os resultados obtidos com o uso desta metodologia não foram bons pois os caminhos não eram diretos, contendo partes desnecessárias, conforme figura 1(a). Porém, com o uso das otimizações descritas neste trabalho, foi possível fazer melhoras significativas nos resultados obtidos sem aumento relevante do tempo de processamento.

8 TRABALHOS FUTUROS

Como trabalhos futuros pretende-se estende o mapeamento de 2D para 3D como fez [10] com o trabalho de [11]. Desta forma será possível o planejamento de trajetória para um dirigível em situações em que haja a necessidade de mudança de altitude, ou até mesmo fazer com que ele voe em uma altitude que acarrete em menor custo.

Outra melhoria que pode ser feita é modificar a função de custo para que a mesma trate a cinemática do dirigível de forma mais realista. Especialmente no ponto que trata das curvas feitas durante o percurso.

Por último, pode-se citar uma extensão para do sistema para realizar o planejamento de trajetória para vários dirigíveis. Fazendo com que estes desviem dos obstáculos e tracem trajetórias que não entrem em estado de colisão, como foi feito em [5] para um sistema de dois robôs.

9 REFERÊNCIAS BIBLIOGRÁFICAS

[1] Baba, C. M.; Corrêa, F. H. M.; Souza, S. A. C.; Wahba, T. M.; Medina, A. F. Otimização da colônia de formigas aplicada ao problema da programação e roteirização de veículos para o transporte de pessoas portadoras de deficiência. Publicado em XXIV Encontro Nacional de Engenharia de Produção. Florianópolis, SC, Brasil. Novembro de 2004. Pág. 2966-2979.

[2] Castro, R. E. Otimização de Estruturas com Multi-Objetivos Via Algoritmos Genéticos. Tese de Doutorado, COPPE/UFRJ. Rio de Janeiro, 2001.

[3] Dorigo M.; Di Caro, G.; Cambardella L. M. (1999). Ant Algorithms for Discrete Optimization. Artificial Life, 5(2):137-172. Also available as Technical Report No. 98-10 (IRIDIA), Université Libre de Bruxelles, Belgium.

[4] Goldbarg, M. c.; Luna, H. P. L. Otimização Combinatória e Programação Linear: Modelos e Algoritmos. Editora Campos, 2ª Edição, 2005.

[5] Jiang, Y.; Zhao, M.; Wang, H.; Fang, L. Hibrid Navigation for a Climbing Robot by Fuzzy Neural Netowrk and Trajectory Planning. Proceedings of the Fourth International Conference on Machine Learning and Cybernetics. Guangzhou, 18-21 August 2005.

[6] Kiguchi, K.; Watanabe, K. Fukuda, T. Trajectory Planning of Mobile Robots Using DNA Compunting. Proceedings of 2001 IEEE International Symposium on Computacional Intelligence in Robotics and Automation. July 29 – August 1, 2001, Banff, Alberta, Canada.

[7] Latombe, J. C. Robot Motion Planning. Kluwer Academic Publishers, Boston, 1991.

[8] Lee, J.; Nam, H. S.; Lyou, J. A Practical Collision-F'ree Trajectory Planning for Two Robot Systems. College of Engineering, Chung-nam University 220 Kung-dong, Yusong-gu, Taejon, 305-764, KOREA.

[9] Pio, J. L. S. ; Campos, M. F. M. . Navegação Robótica. In: SBC - Sociedade Brasileira de Computação. (Org.). Jornadas de Atualização em Informática - Livro Texto. 1 ed. Campinas: SBC - Sociedade Brasileira de Computação, 2003, v. II, p. 389-438.

[10] Sinopoli, B.; Micheli, M.; Donato, G.; Koo, T. J. Vision Based Navigation for an Unmanned Aerial Vehicle. Proceedings of the IEEE International Conference on Robotics an Automation (ICRA 2001). Seoul, Korea. May, 2001.

[11] Thrun, S. Learning metric-topological maps for indoor mobile navigation. Artificial Intelligence, 99(1):21-71, February 1998.