
PROPOSTA DE PLANEJAMENTO DE ROTAS PARA ROBÔS DE INSPEÇÃO USANDO UM ALGORITMO DE COLÔNIA DE FORMIGAS

Roberto Fernandes Tavares Neto ¹ e Leandro dos Santos Coelho ²

¹ LabP3, Rua Iapó, 1245, CEP 80215-020, Curitiba, PR, Brasil, e-mail: roberto@tavares.eng.br

² Pontifícia Universidade Católica do Paraná, PUCPR/CCET, Grupo Produtrônica
Programa de Pós-Graduação em Engenharia de Produção e Sistemas, Laboratório de Automação e Sistemas
Rua Imaculada Conceição, 1155, CEP 80215-901, Curitiba, PR, Brasil, e-mail: lscoelho@rla01.pucpr.br

Resumo: A importância dos ROV's (*Remoted Operated Vehicles* ou veículos operados remotamente) e de robôs autônomos em tarefas de inspeção está cada dia maior. Estas tarefas possuem uma limitação: o tempo necessário para a realização da mesma. Cada momento gasto com a inspeção, gera custos, alguns devido à própria tarefa de inspeção – desgaste do robô/ROV – e custos devido à parada da produção para a inspeção, pois normalmente a execução da inspeção ocorre apenas com o sistema a ser inspecionado desligado, o que causa a perda de possíveis lucros. A inspeção pode ocorrer em ambientes com obstáculos, o que nos gera um problema: normalmente, tais sistemas utilizam um cabo (“cordão umbilical”) entre uma base fixa e o robô. Como este cabo, possui um tamanho máximo fixo, limitando desta forma o movimento do robô. Este artigo descreve uma variação do algoritmo da colônia de formigas para o planejamento da tarefa de inspeção utilizando um robô com a limitação de um cordão umbilical.

Palavras Chaves: robótica, colônia de formigas, serviço, inspeção, veículos operados remotamente.

Abstract: The importance of ROV's (*Remoted Operated Vehicles*) and autonomous robots on inspection tasks are increasing each day. Those tasks have one limitation: the time needed to perform the task. Each moment spent to accomplish the inspection generate costs, some due the inspection task – like the mechanical erode of the robot/ROV – some due the interruption of the production system – the inspection execution can normally occurs only with the target system deactivated, causing lost of possible lucrative production. The inspection can occur in environments with barriers and obstacles, giving us a problem: usually, those systems are provided with a cable between a fixed base and the vehicle. As this cable has a maximum length, we have a limit for the robot action. This paper describes a variation of the ant colony algorithm (ACS) to the inspection task plan using a robot with movements limited by a cable.

Keywords: robotics, ant colony algorithm, service, inspection, remoted operated vehicles.

1 INTRODUÇÃO

A área de robótica industrial tem evoluído rapidamente desde a década passada, devido ao acentuado aumento do poder

computacional e disponibilidade de uma grande variedade de sensores. Isto pode ser observado no fato dos seres humanos terem enviado robôs para Marte, para o fundo dos oceanos, aplicando robôs dentro de reatores nucleares, linhas de manufatura, inspeção de dutos, além da comercialização de brinquedos infantis (brinquedos da Sony, por exemplo, Aibo) e educacionais (por exemplo, kits LEGO e Khepera). Outro indicador deste fato é que as vendas anuais de robôs industriais têm crescido nos Estados Unidos à taxa de aproximadamente 25% ao ano. Além disso, outro aspecto relevante é que os robôs cada vez mais têm sido dotados da capacidade de aprender, atuar autonomamente, e interagir com os humanos e seu ambiente.

Devido à evolução dos meios de produção para o modelo produtivo atual, onde a quantidade de produtos e a rapidez na manufatura são fatores importantes, surge a necessidade de empregados que possam trabalhar da forma mais eficiente para evitar o desperdício, seja ele de tempo ou de recursos materiais. Estes requisitos podem ser atingidos pela robótica, onde se desenvolvem robôs-empregados que auxiliam a produção, sem tempo para descanso e que são aptos a trabalhar em ambientes perigosos, já que “os robôs não necessitam de medidas de segurança e conforto da mesma forma que os seres humanos” (Niku, 2001; Horodincea *et al.*, 2002).

Kopacek (1997) indicou um aumento constante do uso de robôs em aplicações não industriais devido à saturação deste nicho de mercado. De acordo com Kopacek (1997), “Com a estagnação do crescimento para os sistemas robóticos ‘convencionais’, alguns fabricantes de robôs industriais tentam identificar novas áreas de operação.” Kopacek (1997) também menciona que o objetivo desta nova geração de robôs é atuação em “áreas não-convencionais”. Estes robôs podem ser divididos em duas classes principais: (i) as que irão substituir o homem em tarefas que podem comprometer sua integridade física (operação em vácuo, em ambiente submarino, aplicações policiais, entre outros) e (ii) aquelas que têm visam aumentar o conforto do ser humano (como, por exemplo, a ajuda no cuidado de pacientes, guia de cegos, tarefas do lar, entre outros).

Como exemplos de tarefas executadas por robôs, pode-se citar: um robô móvel autônomo para aplicação em agricultura (Mandow *et al.*, 1996), substituindo o trabalho humano em atividades inóspitas, como a pulverização de inseticidas em estufas. Um veículo autônomo especializado em detectar e

socorrer pessoas soterradas em escombros de estruturas danificadas (Murphy, 2000), tal veículo poderia buscar por sobreviventes com um nível de rigor que normalmente é fatigante para os humanos. Neste contexto, os robôs móveis autônomos e cooperativos poderiam trabalhar em equipe para realizar tarefas que seriam impossíveis para humanos e para um único robô (Bay, 1995). Uma aplicação doméstica de robôs móveis é a tarefa de aspirar o pó de aposentos sem intervenção humana.

Nota-se que, em nas aplicações de robôs móveis mencionadas nesta introdução, existe um grande grupo que possui um problema em comum: uma vez que possui um conjunto de tarefas a serem cumpridas, o sistema autônomo deve gerar uma rota que passe por diversos pontos de um mapa, seja o objetivo a entrega de medicamentos em um hospital, seja a inspeção de um sistema de encanamentos.

Um fator importante ao se observar um sistema autônomo que objetiva a obtenção de uma flexibilidade necessária (ou mesmo suficiente) para ter sua lista de tarefas reconfigurada a qualquer instante é que o planejamento do comportamento do robô móvel deve ser realizado com relativa rapidez e melhor otimização possível com menor custo computacional.

Neste contexto, o primeiro problema a ser resolvido é o planejamento da rota a ser seguida pelo robô móvel. Nesta etapa busca-se determinar o trajeto que o robô deve percorrer, evitando ao máximo a redundância e o desperdício de energia — ou seja, otimizando-se a rota com a melhor relação custo *versus* benefício.

Martin (1997) menciona que “*existe uma necessidade do Departamento de Energia (DOE), dos Estados Unidos da América, em inspecionar a condição interna de encanamentos e outros locais inacessíveis, de risco ou de acesso restrito durante descontaminação e descomissionamento de instalações.*”

De acordo com Kunt (1998), “*Mais de 20% dos sistemas de sewerage estão danificados pela idade avançada. A estimativa de custo para a restauração é de aproximadamente 100 milhões de marcos alemães, em 1998. Como consequência, o governo alemão passou a responsabilidade da inspeção dos dutos aos donos, forçando-os a verificar regularmente seu estado através de tecnologias de inspeção cabíveis.*”

Existem muitos outros relatos de problemas semelhantes, em vários países. Observando-se este perfil de mercado, muitas indústrias e centros de pesquisa vêm criando sistemas para a inspeção de tubulação. Diversos sistemas, a exemplo do sistema KARO (Kunt, 1998), que usa ROV's (*remote operated vehicles* — veículos operados remotamente) e já está sendo utilizados em inspeções. Tais sistemas, se compõe de sistemas móveis, dotados de uma ou mais câmeras, sendo controlados por um operador humano através de um sistema fixo na superfície. Os sistemas móvel e fixo são conectados através de um “cabo umbilical”.

Porém, o trabalho a ser realizado de inspeção pode envolver uma rede grande e complexa de caminhos. Para aumentar o problema, o cabo umbilical que se liga ao robô limita a ação do mesmo.

Neste ponto, nota-se que um planejamento prévio do caminho de inspeção a ser realizado pode gerar um aumento de desempenho no procedimento de navegação, planejamento e controle de um robô móvel.

Este problema é abordado, nas próximas seções deste artigo, através de uma modificação do algoritmo da colônia de formigas (ACO — otimização da colônia de formigas, *ant colony optimization*) proposto por Dorigo *et al.* (1996). É interessante notar que a alteração de tal algoritmo visa a obtenção de uma resposta promissora, com maior flexibilidade e robutez, ao planejamento das rotas a serem percorridas pelo robô. Neste caso, em primeira instância são abordadas uma série de premissas aplicadas à inspeção. Estas premissas são contextualizadas de forma a, caso o problema mude de forma — por exemplo, mude-se de inspeção para aplicações de entrega de material — o mesmo conceito possa ser aplicado, mudando apenas as referências do robô móvel.

2 ALGORITMO DE OTIMIZAÇÃO POR COLÔNIA DE FORMIGAS

O algoritmo de otimização por colônia de formigas, é uma forma de otimização já utilizada em problemas de múltiplas rotas. Dorigo & Di Caro (1997) inicia o estudo da implementação do comportamento de formigas reais em situações aonde se busca determinar o menor caminho entre dois pontos. De acordo com os autores, “as formigas reais são capazes de descobrir o menor caminho entre uma fonte de comida e seu ninho, sem o uso de pistas visuais e sim com o uso de informação relativa ao feromônio”. O algoritmo consiste da utilização de agentes reativos que se comunicam com o meio ambiente através de “feromônios” para otimização de uma determinada tarefa, geralmente uma tarefa de percorrer um grafo. Este tipo de comunicação pode ser visto como sendo um modelo de comunicação sinérgica.

Este grafo, é definido por vários pontos (objetivos), ligados entre si por múltiplos caminhos. Quando um agente (“formiga”) passa por um caminho, ele incrementa uma determinada quantidade de feromônio (ou “peso”). Quando todos os agentes cumprem seus objetivos, o feromônio dos caminhos é atualizado, tanto no aumento de feromônio (ao se computar os acréscimos), quanto na redução do mesmo (visando assim simular a “evaporação” do feromônio).

A inteligência da colônia se dá no momento em que uma “formiga” decide entre os vários caminhos possíveis. Dorigo & Gambardella (1997) utilizam uma regra aleatória-proporcional para esta decisão, baseada em uma taxa de proporcionalidade entre os feromônios atuantes. Assim, os agentes em maioria escolhem os caminhos já percorridos em menor tempo. Entretanto, principalmente no início do processo iterativo — quando as diferenças de feromônios são menores — um agente que saia do “caminho padrão” pode descobrir uma melhor resposta para o problema.

Esta abordagem foi apresentada para a resolução do problema do caixeiro viajante (Bianchi *et al.*, 2002). Este problema consiste na determinação da melhor forma de um vendedor passar entre várias cidades (ver figura 1). Para tal problema, o algoritmo de colônia de formigas pode ser aplicado, conforme demonstraram Bianchi *et al.* (2002), Dorigo & Gambardella (1997), entre outros.

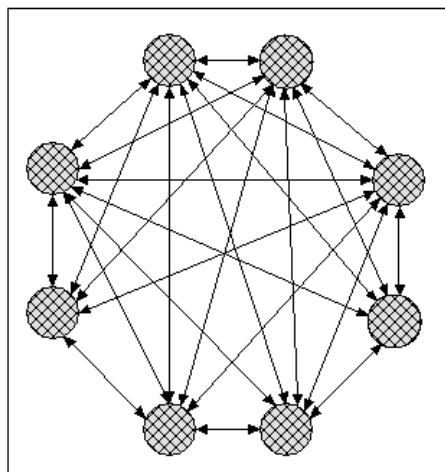


Figura 1. O problema clássico do caixeiro viajante.

2.1 Implementação da Colônia de Formigas

O pseudo-código aqui mostrado é baseado no algoritmo mostrado por Dorigo & Di Caro (1999), que descreve o funcionamento clássico de um algoritmo de colônia de formigas.

O ACO mostra a inteligência da resolução de um problema combinatório como emergente da interação entre agentes através do meio. Assim, a inteligência não surge de um agente isolado, mas da comunicação entre cada agente individual e o meio. E é através da interação com o meio que os agentes conseguem obter uma solução otimizada para o problema.

Esta interação ocorre através da diferença do feromônio no meio. Ao percorrer um caminho, a formiga deposita uma certa quantidade de feromônio.

Quando a formiga se encontra em uma encruzilhada, ela realiza uma escolha ponderada. Tal escolha é realizada atribuindo um peso à cada probabilidade de cada caminho proporcional ao percentual de feromônios presente em todos os caminhos adjacentes.

As principais modificações realizadas no AÇO objetivam contribuir para obtenção de um algoritmo mais simples e flexível de forma a estabelecer algumas condições de contorno. O resultado destas alterações é apresentado na figura 2. Até o atual estágio de desenvolvimento e implementação da proposta apresentada neste artigo, as seguintes modificações foram realizadas no algoritmo apresentado por Dorigo & Di Caro (1999):

- determina-se que o critério de parada está vinculado ao número de agentes criados (linha 2). O objetivo é juntamente com o critério mostrado na linha 11, conseguir facilmente se determinar o tempo máximo de operação do algoritmo;
- não utiliza a premissa de que todos os caminhos estão disponíveis: ao invés disso, são geradas três categorias de caminhos possíveis: caminhos ainda não passados, caminhos já passados e caminhos proibidos.
- para objetivos futuros deste projeto, adota-se o inverso do que normalmente se estabelece como meta: não se busca, através deste algoritmo, o melhor caminho para visitar todos os pontos de intersecção (lógica do caixeiro-viajante), mas sim passar pelos caminhos permitidos.

3 ANÁLISE DO PROBLEMA DE INSPEÇÃO

Neste trabalho, se definiu o problema de inspeção em tubulações com as seguintes características (ver figura 3):

```

01: Meta-Heurística da colônia de formigas {
02:   Enquanto (o número de formigas for menor que o
03: estabelecido) {
04:     Gere_mais_uma_formiga
05:     Compute_mudanças_do_feromônio
06:   }
07: }

08: Procedimento Gere_mais_uma_formiga {
09:   Crie uma formiga no ninho
10:   Enquanto ((formiga não percorreu todo o trajeto) e
11:             (formiga está andando por um tempo
12: menor que o tempo máximo)) {
13:     Force_movimento_da_formiga }
14:   Avalie_caminho
15: }

16: Procedimento Force_movimento_da_formiga {
17:   Escolha_melhor_caminho
18:   Mova_para_melhor_caminho
19: }

20: Procedimento Escolha_melhor_caminho {
21:   - Tome como caminhos possíveis todos os
22: caminhos existentes ligados na localização atual da
23: formiga com exceção ao caminho de onde a formiga
24: veio e os caminhos proibidos.

25:   - Escolha-os de forma aleatória, porém variando as
26: probabilidades de cada caminho, garantindo que a
27: proporcionalidade entre a chance de escolha de um
28: caminho e a razão entre a quantidade de feromônio
29: existente e a quantidade de todos os caminhos seja
30: mantida.
31: }

32: Procedimento Mova_para_melhor_caminho {
33:   Coloque uma unidade de feromônio no caminho atual
34:   Mova a formiga para o melhor caminho
35: }

36: Procedimento Compute_mudanças_do_feromônio {
37:   Seguindo uma regra de evaporação de feromônio,
38: diminua a quantidade do mesmo em cada caminho.
39: }

```

Figura 2. Pseudo-código do algoritmo de colônia de formigas.

- o problema se reduz a um mapa aonde todos os caminhos devem ser percorridos no menor tempo possível;
- desta forma, os caminhos existentes são conhecidos *a priori*;

- necessariamente vão existir caminhos que o robô não pode passar (proibidos), devido à problemas físicos (encanamentos quebrados, caminhos bloqueados, etc).

Como simplificação do problema, assumiu-se que a distância entre duas intersecções é sempre constante.

Existe ainda outro problema a ser tratado: a limitação do robô que irá percorrer os caminhos. No caso deste artigo, tomou-se como base o ROV (*Remoted Operated Vehicle* ou veículo operado remotamente) do fabricante INUKTUN, utilizado por pesquisadores do CENPRA, de Campinas-SP, para a inspeção em galerias. Este ROV, possui um centro de operação fixo em terra, aonde um operador humano o controla. A ligação entre este centro de operação e o ROV é realizada através de um cabo umbilical, que provém os sinais de controle e de alimentação para o veículo. Assim, tornou-se claro que existe uma restrição importante a ser considerada: o tamanho do cabo.

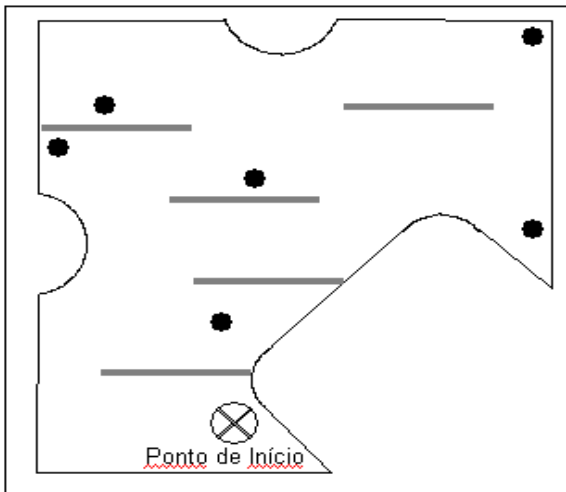


Figura 3. Um exemplo de problema de inspeção. Neste caso, o robô inicia o movimento no ponto indicado e deve passar pelos pontos pretos. No entanto, existem barreiras (em cinza) que dificultam o movimento do mesmo.

Esta característica foi incluída no sistema através do pseudo-código mostrado na figura 4. Nota-se que agora se tem a referência a uma entidade denominada pilha_cabo. Este elemento é uma fila FILO (*first in, last out*, primeiro a sair é o último a entrar) convencional, aonde é determinada a quantidade gasta de cabo. A análise deste elemento mostra que ele deve ser suficientemente estruturado para que seja possível realizar as seguintes operações:

- adição e remoção dos elementos da pilha FILO;
- visualização do último elemento inserido;
- obtenção do tamanho da fila;
- e, conseqüentemente, tamanho do cabo gasto: Como se definiu que o espaçamento entre os pontos de intersecção é constante, e cada elemento da fila representa uma distância entre dois pontos. O tamanho do cabo umbilical pode ser facilmente expresso em função do número de elementos inseridos na pilha.

```

01: Procedimento Mova_para_melhor_caminho {
02:   Coloque uma unidade de feromônio no caminho atual
03:   Mova a formiga para o melhor caminho
04:   Se o caminho atual for o último caminho
05:     da pilha_cabo, remova-o da mesma. Caso contrário,
06:     o adicione.
07: }
08:
09: Procedimento Escolha_melhor_caminho {
10:   -Se o cabo está esticado, o único caminho possível
11:   é a volta por onde veio.
12:   -Se o cabo não está esticado, tome como caminhos
13:   possíveis todos os caminhos existentes ligados
14:   na localização atual da formiga com exceção
15:   ao caminho de onde a formiga veio e os proibidos.
16:   - Escolha-os de forma aleatória, porém variando as
17:   probabilidades de cada caminho, garantindo que a
18:   proporcionalidade entre a chance de escolha de um
19:   caminho e a razão entre a quantidade de feromônio
20:   existente e a quantidade de todos os caminhos seja
21:   mantida.
22: }

```

Figura 4. Modificação do ACO para a tarefa de inspeção do robô móvel com cordão umbilical.

4 MODELAGEM DO PROBLEMA

Para verificar o conceito que foi descrito nas seções anteriores utilizou-se de uma implementação em JAVA. Esta implementação se baseou em duas classes principais:

- uma classe representando o meio ambiente, que além do mapeamento do meio, armazena durante a execução do programa informações relativas ao feromônio das formigas;
- uma classe representando a formiga. As instâncias desta classe possuem uma referência para o objeto que representa o meio, e lá depositam o feromônio. Assim, não há a comunicação entre formigas, apenas entre o meio e a formiga, garantindo assim uma comunicação sinérgica.

Inicialmente, se definiu a atualização do feromônio através das equações apresentadas por Dorigo & Di Caro (1999). Visando simplificar a concepção do algoritmo de colônia de formigas, posteriormente tentou-se, com sucesso, linearizar o problema, conforme as seguintes regras:

- a adição do feromônio: independente da quantidade de feromônio existente, a formiga sempre adiciona uma unidade de feromônio na trilha quando for conveniente; e
- a evaporação do feromônio: também buscou-se uma simplificação. Ao contrário da evaporação do feromônio descrita em Dorigo & Di Caro (1999), neste caso, sempre se tratou a evaporação como sendo a uma taxa constante de 10% do feromônio depositado no caminho.

O objetivo destas simplificações é possibilitar posteriormente a implementação deste mesmo algoritmo em sistemas de baixo poder computacional, estes comuns em sistemas de controle de robôs, a exemplo de microcontroladores da linha PIC ou 8051.

Para a escolha da rota, utilizou-se o mesmo procedimento descrito por Dorigo *et al.* (1997) e Dorigo & Di Caro (1999). O

algoritmo implementado é descrito no pseudo-código da figura 5.

```

01: Obtenha a soma de todos os feromônios de todos os
    caminhos permitidos
02: Sorteie um número entre 0 e 1, guarde em Rand
03: Enquanto Rand > 0 {
04:     Pegue o próximo caminho
05:     Faça Rand = Rand - ((quantidade de feromônio do
06: caminho atual) / (quantidade total de feromônio))
07: }
08: O caminho atual é o escolhido

```

Figura 5. Procedimento de escolha de rota em intersecções.

5 RESULTADOS OBTIDOS

O algoritmo apresentado anteriormente foi implementado em JAVA. Foi-se proposto um grafo de ensaio com 11 nós, conforme mostrado na figura 6.

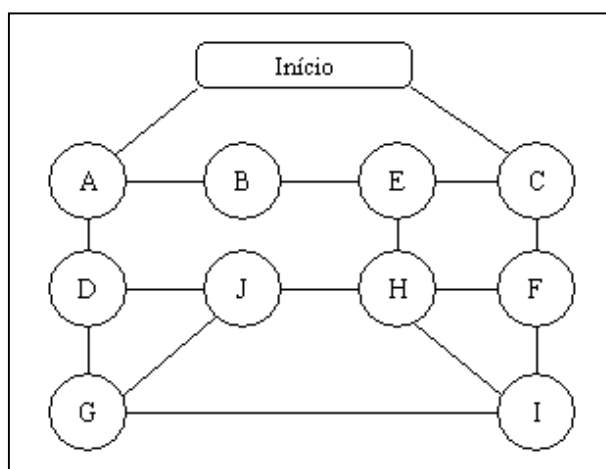


Figura 6. O grafo de teste.

Selecionou-se como objetivo a passagem pelos nós D e H, e com uma limitação devido à corda de 10 movimentos. O resultado é mostrado na figura 7.

```

0.[C, INICIO, C, F, H, I, G, D]
1.[A, D, J, H]
2.[A, B, A, D, I, H]
3.[C, E, H, I, G, D]
4.[C, F, H, I, G, D]
5.[C, F, C, E, H, F, C, E, H, I, G, I, H, E, C, INICIO, A, D]
6.[A, D, A, INICIO, C, F, H]
7.[A, INICIO, A, D, I, H]
8.[A, INICIO, C, F, H, I, G, D]
9.[A, B, A, D, I, H]
10.[C, INICIO, C, E, H, I, G, D]
11.[A, B, A, INICIO, C, E, H, F, C, E, H, E, C, INICIO, A,
INICIO, C, F, H, E, C, INICIO, A, B, E, H, I, G, D]
12.[C, F, H, E, C, INICIO, A, B, E, B, A, INICIO, C, F, H, I, G,
I, H, E, C, E, H, F, C, E, H, I, G, D]
13.[A, D, A, INICIO, C, F, H]
14.[A, D, A, B, E, H]
15.[A, B, A, INICIO, C, F, H, I, G, D]
16.[C, INICIO, C, E, H, F, C, INICIO, A, D]
17.[A, D, A, INICIO, C, F, H]
18.[C, F, C, E, H, I, G, D]
19.[A, D, A, B, E, H]
20.[A, D, A, B, C, F]
21.[A, D, J, H]

```

Figura 7. O resultado das iterações do ACO proposto.

Quando existe um retorno ao nó anterior (marcado em negrito), nota-se a ocorrência do retorno imposto pelo cordão umbilical.

O cordão umbilical torna vários caminhos proibitivos devido ao retorno que causa na trajetória da formiga. Quando, mesmo tendo o cordão umbilical estendido, a formiga consegue atingir seu objetivo, o traço de feromônio é estendido para mais nós, causando um distúrbio no meio. Porém, conseguiu-se determinar um caminho ótimo para o mesmo.

6 CONSIDERAÇÕES FINAIS

Neste artigo, mostrou-se que, o algoritmo de colônia de formigas com algumas modificações pode ser usado para o planejamento de rotas de inspeções por robôs dotados de cordão umbilical em ambiente com a presença de obstáculos.

Para uma maior adequação ao contexto do problema, sugeriu-se simplificações no algoritmo, permitindo assim a consistência do mesmo ao âmbito dos sistemas autônomos baseados em sistemas processados com baixo poder computacional.

Espera-se que este seja um passo em direção à transformação dos ROV's de inspeção em verdadeiros robôs autônomos, buscando sempre uma otimização dos procedimentos de inspeção automatizada.

Além disso, nos trabalhos futuros a serem desenvolvidos, neste contexto, deseja-se apresentar um estudo de formas de cooperação entre robôs móveis (multi-agentes) em tarefas realizadas em ambientes que apresentam mudanças dinâmicas e que possuem fatores com a presença de incerteza.

REFERÊNCIAS BIBLIOGRÁFICAS

- Bay, J. S. (1995). Design of the army-ant cooperative lifting robot, *IEEE Robotics & Automation Magazine*, vol. 2, no. 1, pp. 36-43.
- Bianchi, L., Gambardella, L. M., Dorigo, M. (2002). An ant colony optimization approach to the probabilistic traveling salesman problem, *Proceedings of PPSN-VII, Seventh International Conference on Parallel Problem Solving from Nature*, Lecture Notes in Computer Science, Springer Verlag, Berlin, Germany, pp. 883-892.
- Dorigo M.; Di Caro, G. (1999). The ant colony optimization meta-heuristic, In D. Corne, M. Dorigo and F. Glover (editors), *New Ideas in Optimization*, McGraw-Hill, pp. 11-32.
- Dorigo M.; Gambardella, L. M. (1997). Ant colonies for the traveling salesman problem, *BioSystems*, vol. 43, pp. 73-81.
- Dorigo, M.; Maniezzo, V.; Colomni, A. (1996). The ant system: optimization by a colony of cooperating agents, *IEEE Transactions on Systems, Man, and Cybernetics — Part B: Cybernetics*, vol. 26, no. 1, pp. 1-13.
- Horodinca, M.; Doroftei, I.; Mignon, E.; Preumont, A. (2002). A simple architecture for in-pipe inspection robots. *International Colloquium on Mobile and Autonomous Systems*, 10 Years of the Fraunhofer IFF, Magdeburg.
- Kopacek P.; Kronreif G.; Probst R. (1997). Modular service robots state of the art and future trends, *Proceedings of International Conference on Advanced Robotics*, Monterey, USA, pp. 51-56.
- Kuntze, H. B.; Karlsruhe, H. (1998), Experiences with the development of a robot for smart multisensoric pipe inspection, *Proceedings of the IEEE International*

Conference on Robotics & Automation, Leuven, Belgium, vol. 2, pp. 1773-1778.

Mandow, A.; Gomes-De-Gabriel, J. M.; Martinez, J. L.; Muñoz, V. F.; Ollero, A.; García-Cerezo, A. (1996). The autonomous mobile robot aurora for greenhouse operation, *IEEE Robotics and Autonomous Magazine*, vol. 3, no. 4, pp. 18-28.

Martin, L. (1997). Multisensor inspection & characterization robot for small pipes (Microspi), *Technology Development Data Sheet*, Federal Energy Technology Center, pp. 161-162.

Murphy, R. R. (2000). Marsupial and shape-shifting robots for urban search and rescue, *IEEE Intelligent Systems*, vol. 15, no. 2, pp. 14-19.

Niku, S. B. (2001). *Introduction to robotics analysis, systems, and applications*, Prentice Hall, New Jersey, USA.