

ABORDAGEM DE BUSCA EM TEMPO REAL PARA MISSÕES DE PATRULHAMENTO COOPERATIVO COM MÚLTIPLOS VANTs

KRISTOFER STIFT KAPPEL¹; TAUÃ CABREIRA MILECH²; LISANE BRISOLARA DE BRISOLARA³; PAULO ROBERTO FERREIRA JÚNIOR⁴

¹Universidade Federal de Pelotas – kskappel@inf.ufpel.edu.br

²Universidade Federal de Pelotas – tmcabreira@inf.ufpel.edu.br

³Universidade Federal de Pelotas – lisane@inf.ufpel.edu.br

⁴Universidade Federal de Pelotas – paulo@inf.ufpel.edu.br

1. INTRODUÇÃO

Os Veículos Aéreos Não Tripulados (VANTs) tem sido extensamente usados em diversas tarefas, tais como, missões de busca e resgate (NATTERO et al., 2014), vigilância de incêndios florestais (CASBEER et al., 2006), detecção de minas terrestres (CASTIBLANCO et al.), dentre outras. Muitas dessas missões estão relacionadas com problemas como cobertura de terrenos e planejamento de rotas. Muitas das abordagens para esses problemas são movimentos geométricos como o boustrophedon (zigzag) ou movimentos circulares, porém esses movimentos não são eficientes em missões de patrulhamento ou de rastreamento, além de serem previsíveis.

Um grupo de VANTs podem realizar as tarefas citadas em menor tempo, uma vez que é possível dividir a área em quadrantes e cada VANT operar sobre um quadrante. No entanto, a melhor solução completa é classificada como NP-Difícil.

Métodos de busca em tempo real usam estratégias baseadas na maneira como formiga se guiam por locais, utilizando valores para simular o feromônio deixado.

Além disso, em cenários reais os veículos deveriam minimizar o número de curvas para minimizar o consumo e maximizar o tempo da missão.

2. METODOLOGIA

Nesse trabalho utilizamos alguns algoritmos citados constantemente na literatura sobre o assunto, como o Node Counting (NC) (PIRZADEH; SNYDER, 1990), Learning Real Time A* (LRTA*) (KORF, 1990), Thrun's Value-Update Rule (TVUR) (THRUN, 1992), Wagner's Value-Update Rule (WVUR) (WAGNER et al., 1997), os quais tem baixo custo computacional e podem ser usados em veículos com processamento limitado.

O cenário é dividido em pequenas células e cada veículo só consegue ver as células vizinhas imediatamente ortogonais em relação a sua. Cada uma destas células, tem um valor associado, chamado de *u-value*, o qual simula o feromônio deixado pelo movimento dos outros veículos no mapa.

O veículo anda pelo cenário sempre escolhendo a célula vizinha com o menor *u-value* e se movendo para ela, o que muda entre os algoritmos é a maneira com o qual o *u-value* é atualizado.

Primeiramente propomos o *Centralized NC-Drone* (CNC-D), que busca consumir menos energia em missões de patrulhamento coletivas. Após, propomos um *Decentralized NC-Drone* (DNC-D), onde os veículos usam uma matriz interna para guardar os locais visitados ao invés da estratégia de feromônios.

Fazer curvas aumenta o consumo de energia dos drones, uma vez que eles precisam desacelerar, mudar a direção e acelerar novamente, o CNC-D busca reduzir isso, fazendo com que sempre que haja um empate no *u-value* das

células vizinhas ele prefira ir para frente, ao invés da escolha aleatória que era realizada anteriormente.

Geralmente, tarefas de patrulhamento com múltiplos VANTs são feitos com o apoio de uma *Ground Control Station* (GCS), o qual centraliza as informações em uma estação, essa estação se comunica com os VANTs enviando dados, como por exemplo, para qual direção o drone deverá se mover e também recebe informações sobre a posição para simular o efeito dos feromônios. Porém, esse método está muito sujeito a falhas, uma vez que os veículos dependem exclusivamente do GCS, a perda de sinal pode comprometer a tarefa e provocar acidentes.

No DNC-D proposto, cada veículo mantém um mapa interno do ambiente na forma de uma matriz, assim, ele salva o número de vezes que passou em cada local, esses dados são atualizados quando ele entra em um certo alcance de outro drone, assim os dois trocam informações e atualizam suas matrizes internas, nós avaliamos três maneiras diferentes de atualizar a matriz, o valor máximo, que compara cada posição em ambas as matrizes e escolhe o maior valor (MAX), a média dos valores (AVERAGE) e um terceiro método, que também escolhe o maior valor, mas assim que há uma cobertura completa a matrix é preenchida novamente com zeros (RESET).

Nos testes, além dos algoritmos já citados, testamos também os métodos geométricos (boustrophedon e circular) e um DNC-D sem nenhuma comunicação entre os veículos.

Todos os algoritmos foram implementados e simulados usando um ambiente de sistemas multiagentes, o Netlogo. O cenário é um *grid* de tamanho 50 x 50 com 4 veículos simultâneos. As simulações foram de 30 rodadas com 5,000, 10,000, 15,000 e 20,000 *ticks* (medida de tempo do netlogo).

As métricas utilizadas para medir a performance dos algoritmos foram, Média Quadrática dos Intervalos (MQI), que avalia a regularidade das visitas em um local, Desvio Padrão das Frequências (DPF), que avalia a uniformidade das visitas, Número de Manobras de Curva (NMC) e Número de Coberturas Completas (NCC).

3. RESULTADOS E DISCUSSÃO

Todos os dados citados adiante podem ser conferidos nas tabelas no seguinte link: <https://imgur.com/a/NbyJG>. Primeiro comparamos o CNC-D com os quatro algoritmos de busca em tempo real citados anteriormente. Em relação ao MQI, Com 5,000 ticks, não há nenhuma diferença estatisticamente relevante, porém a partir de 10,000 ticks o CNC-D começa a se sobressair à todos os outros métodos. Em 20,000 ticks o CNC-D supera o TVUR, que é a segunda melhor métrica, em 5% e o NC original em 7%.

Os resultados dos mesmos algoritmos comparados pelo DPF apontam que o CNC-D, em 5,000, obtém resultados melhores do que os do TVUR apenas. A partir de 10,000 a 20,000 ticks ele é melhor que o NC, LRTA* e o TVUR.

Como os métodos de busca em tempo real tomam decisões aleatórias em caso de empate eles tendem a produzir resultados mais instáveis, podendo concentrar a navegação em uma região do ambiente por um longo tempo e como o DPF e o MQI medem regularidade, quanto mais ticks mais o CNC-D se sobressai em relação aos outros algoritmos.

Como o esperado, o CNC-D se sai melhor do que os algoritmos de busca em tempo real quando comparado a média do NMC, obtendo resultados três ou quatro vezes menores.

Após, comparamos o DNC-D Sem Comunicação contra os métodos geométricos, ou seja, o boustrophedon motion (zigzag) e o circular. Nos métodos geométricos o cenário é dividido em quatro quadrantes e cada um desses é designado a um veículo, ou seja, cada veículo realiza o movimento pré-definido em um quarto do ambiente.

Quanto ao MQI, o boustrophedon apresenta o melhor resultado sempre, já em relação ao DPF e o NCC, o circular é o melhor, porém o DNC-D tem o menor NMC.

Por último, comparamos os tipos de sincronização do DNC-D, com o DNC-D Sem Comunicação e o CNC-D. No MQI, o CNC-D é a melhor solução, pois os veículos tem todas as informações do cenário, o DNC-D Sem Comunicação tem os piores resultados, pois tem visão da sua própria matriz apenas e não trocam informações em nenhum momento, já o MAX e o AVERAGE tem resultados melhores do que o RESET.

No DPF o CNC-D e o RESET tem os melhores e piores resultados respectivamente, nessa métrica o DNC-D Sem Comunicação supera qualquer versão com comunicação. No NMC, o DNC-D Sem Comunicação é o melhor método, pois, como os drones não trocam informações eles enfrentam mais situações em que podem ir reto ao invés de fazerem curvas. Por final, no NCC, o CNC-D é o que apresenta os melhores resultados, seguido pelo DNC-D Sem Comunicação, MAX, AVERAGE, RESET.

4. CONCLUSÕES

Este trabalho apresenta uma variação do Node Counting com preocupação energética, o NC-Drone, que, em caso de empate, prefere sempre seguir reto ao invés de realizar uma curva.

O NC-Drone com acesso às informações do ambientes foi melhor do que todas as estratégias de busca em tempo real na MQI, foi melhor do que muitos métodos e similar a outros no DPF e no NCC, porém, diminuiu drasticamente o número de curvas comparados com as outras heurísticas.

Também propomos o NC-Drone descentralizado, ou seja, cada um guardando suas informações do mapa e maneiras para se comunicar com os outros veículos no cenário para atualizar as informações guardadas. Quando comparamos a versão sem comunicação deste algoritmo com os métodos tradicionais geométricos notamos que ele perde para o circular no geral mas se sobressai em relação ao zigzag na maioria das métricas.

Por último, considerando apenas as variações do NC-Drone, podemos concluir que as versões descentralizadas emergem como estratégias promissoras para tarefas de cobertura, uma vez que tem bom desempenho em métricas relacionadas a patrulhamento. A versão centralizada tem os melhores resultados, menos no NMC, mas essa versão não é viável no mundo real pela constante comunicação entre os veículos e a base. Nas versões descentralizadas, o NC-Drone Sem Comunicação apresenta os melhores resultados.

No futuro vamos buscar melhorar os mecanismos de sincronização, implementar os novos métodos em missões reais usando um multicóptero. Com nosso trabalho atual e futuros trabalho buscamos avançar o estado da arte em estratégias de planejamento de missões viáveis no mundo real para veículos aéreos autônomos



5. REFERÊNCIAS BIBLIOGRÁFICAS

NATTERO, C. et al. Coverage Algorithms for Search And Rescue with UAV Drones. **Artificial Intelligence, Workshop of the XIII AI*IA Symposium**, 2014.

CASBEER, David W. et al. Cooperative forest fire surveillance using a team of small unmanned air vehicles. **International Journal of Systems Science**, v.37, n. 6, p. 351 – 360, 2006.

CASTIBLANCO, C. et al. Air Drones for Explosive Landmines Detection. In: **ROBOT2013: First Iberian Robotics Conference**. Springer International Publishing, Vol.2 , p. 107 – 114, 2014.

PIRZADEH, Amir; SNYDE, Wesley. A unified solution to coverage and search in explored and unexplored terrains using indirect control. In: **Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on**. IEEE, p. 2113–2119, 1990.

KORF, R. Real-time heuristic search. **Artificial intelligence** v. 42, n. 2-3, p. 189–211, 1990

THRUN, Sebastian B. Efficient exploration in reinforcement learning. 1992.

WAGNER, I.; LINDENBAUM, Michael; BRUCKSTEIN, A. On-line graph searching by a smell-oriented vertex process. In: **Proceedings of the AAAI Workshop on OnLine Search**. p. 122–125, 1997.